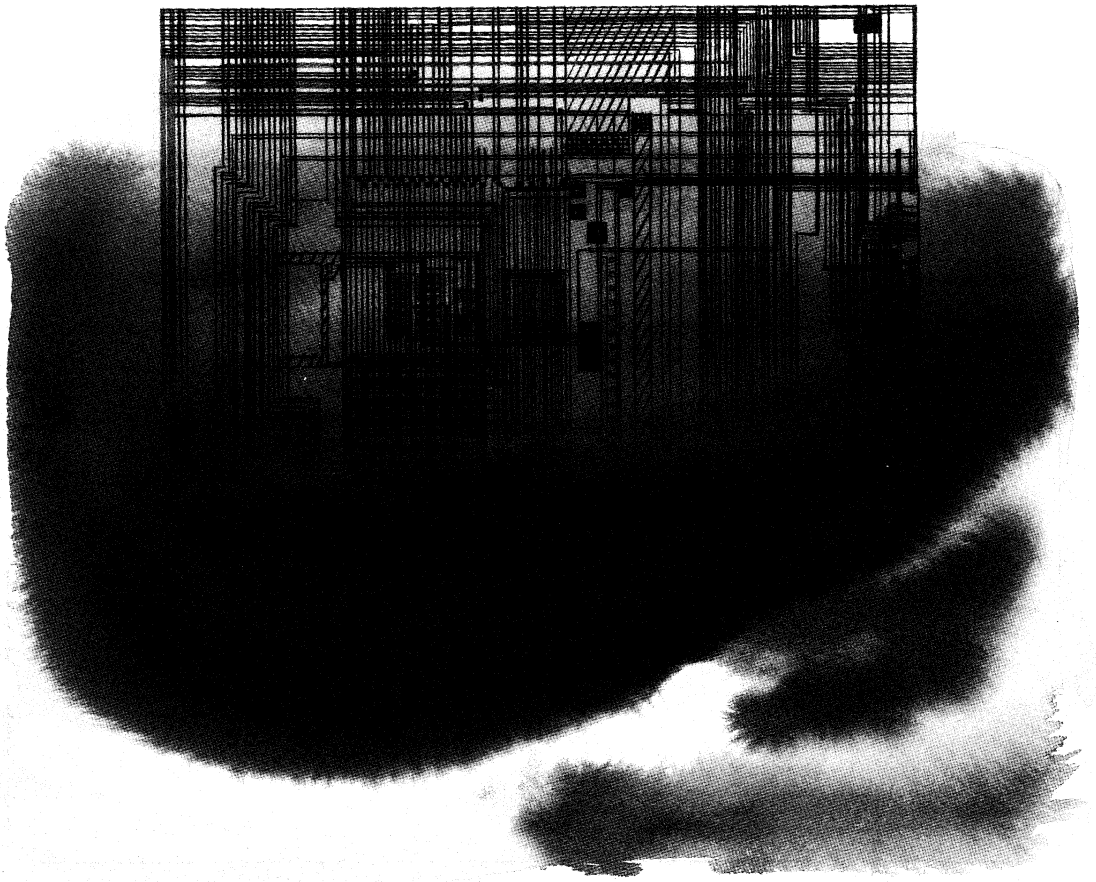
XILINX®

The Programmable Logic Data Book



1994

The Programmable Logic Data Book

2100 Logic Drive
San Jose, California 95124
Telephone: (408) 559-7778

XILINX®

SECTION TITLES

XILINX®

TABLE OF CONTENTS

TABLE OF CONTENTS


XILINX
XACT
DEVELOPMENT SYSTEM SOFTWARE

XILINX®

Programmable Logic Devices

About the Company...

Xilinx was founded in 1984, based on the revolutionary idea to combine the logic density and versatility of gate arrays with the time-to-market advantages and off-the-shelf availability of user-programmable standard parts. One year later, Xilinx introduced the world's first Field-Programmable Gate Array (FPGA). Since then, the company has continually improved device densities and speeds, while lowering costs. In fact, over the last eight years, Xilinx devices boasted a 35%-per-year improvement in speed, a 55%-per-year increase in density and a 46%-per-year decrease in cost per function.

In 1992, Xilinx expanded its product line to include advanced EPLDs (EPROM technology-based complex Programmable Logic Devices). The company has since added EPLDs designed for high performance and high density on the same chip. For the user, EPLDs are an attractive complement to FPGAs, offering simpler software and more predictable timing.

As the market leader in the fastest-growing segment of the semiconductor industry, Xilinx strategy is to focus its resources on creating new ICs and development system software, on developing markets, and on building a diverse customer base across a broad range of geographic and market-application segments. The company avoids the large capital commitment and overhead burden associated with owning a wafer fabrication facility by establishing manufacturing alliances with several high-volume state-of-the-art CMOS memory manufacturers. Using standard high-volume memory processes assures lowest manufacturing cost, produces programmable logic devices with well-established reliability, and provides for an early access to advances in CMOS technology.

The company markets its products in North America through a network of seven direct-sales offices, manufacturers' representatives in 75 locations, as well as five distributors. Outside North America, the company sells its products through direct-sales offices in England, Germany, Japan, and Hong Kong, and through representatives and distributors in 36 countries.

With 1993 revenues of over $230 million, Xilinx is the world's largest supplier of CMOS programmable logic and the market leader in Field Programmable Gate Arrays.

One Source for FPGAs and EPLDs

For designers most comfortable with the speed, design simplicity, and predictability of PALs, the XC7000 Family of complex EPLDs provide a higher level of integration, with the *same familiar* PALASM and ABEL design methodology.

For a move up to higher density designs that combine an abundance of gates and I/Os with fast system speed, Xilinx offers the ideal logic device, the FPGA: Three complete families with over 30 different devices, including the world's largest FPGA, the 25,000-gate XC4025. There are more than 500 product types, plus more than 40 varieties of devices for military and aerospace applications. Design software is also available that is fully integrated, highly automated, easy to use, and works with existing CAE tools.

Programmable Logic vs Gate Arrays

Faster Design and Verification – Xilinx FPGAs and EPLDs can be designed and verified in a few days, while the same process requires several weeks with gate arrays. There are no non-recurring engineering (NRE) costs and no prototypes to wait for.

Design Changes without Penalty – Because the devices are software configured via instant programming, modifications are much less risky and can be made anytime, in a matter of hours instead of the weeks it would take with a gate array. This adds up to significant cost savings in design and production.

Shortest Time to Market – Designing with Xilinx programmable logic vs gate arrays, time-to-market is measured in days or a few weeks, rather than the months required when designing with gate arrays.

A study by McKinsey & Co. concludes that a six-month delay in getting to market can cost a product *one-third of its lifetime potential profit*. With a custom gate array, design iterations can easily add that much time, and more, to a product schedule.

Once the decision has been made to use Xilinx programmable logic, a choice must be made from a number of *product families, device options,* and *product types*. The following guides simplify the selection process.

Selecting the Right Device

Step 1 - Choose a Family

The Family Architecture Comparison and Speed and Density charts help you determine whether an XC7000 series EPLD, XC2000/XC3000/XC3100 Series FPGA or XC4000/XC4000A/XC4000H Series FPGA is right for your application. Comparative information is provided on product architecture, logic capacity, design timing, system features, etc.

Step 2 - Choose a Device

Now that you've determined which Family of Xilinx products works best for you, use the *Product Comparison* chart to select specific device(s) within the Family. Comparisons are provided for gate-count, number of I/Os, flip-flops, RAM bits, CLBs and Macrocells.

Step 3 - Choose a Package

Finally, the charts entitled *Package Options and I/O Pins Per Package* show the 300+ package/speed/temperature and qualification level options Xilinx offers. Since many products come in common packages with common footprints, designs can often be migrated to higher or lower density devices without any board changes.

Family Architecture Comparison

	EPLDs		FPGAs	
	XC7200 Family	**XC7300 Family**	**XC2000/L XC3000/A/L XC3100 Family**	**XC4000/A/H Family**
Architecture	PAL-like, AND-OR plane Macrocells and product terms	Advanced PLD – high speed, high density function blocks (FB) in the same device	Gate array – like Many small blocks	Gate array – like Many small blocks
Logic Capacity	36 – 72 macrocells Integrate 4 – 8 PAL/ 22V10s	36 – 144 macrocells Integrate 4 – 16 PAL/ 22V10s	600 – 7,500 gates Integrate TTL, MSI, PLDs	1,600 – 25,000 + gates Integrate TTL, MSI, PLDs, RAM
Design Timing	Fixed, PAL-like 60 MHz – predictable	Fixed, PAL-like 100 MHz – predictable	Gate array-like – depends on application Can be >100 MHz, typically 25 – 40 MHz (XC3000) or 50 – 80 MHz (XC3100)	Gate array-like – depends on application Can be >100 MHz, typically 30 – 50 MHz
Number of I/Os	36 – 72	38 – 156	Many – like gate array 58 – 176	Many – like gate array 64 – 192
Number of FF	72 – 144	36 – 234	Many – like gate array 122 – 1,320	Very large number - 256 – 1,536 plus on-chip RAM - up to 18.4 K bits
Power Consumption	0.5 – 1.25 W static 0.75 – 1.5 W typical	0.4 – 2.0 W static 0.5 – 2.25 W typical Programmable power management	Very low, mW static Dynamic – depends on application 0.12 – 1.0 W typical	Very low, mW static Dynamic – depends on application 0.25 – 2.0 W typical
System Features	100% interconnect guaranteed Arithmetic carry logic ALU per macrocell 3.3 V/5 V I/O capability for XC7200/XC7300	100% interconnect guaranteed Arithmetic carry logic ALU per macrocell 3.3 V/5 V I/O capability for XC7200/XC7300 3 global clocks 12 mA/24 mA output drive Carry look ahead High output drive	Two global clock buffers Programmable output slew rate Internal 3-state busses Power-down mode 8 mA output drive for XC3100	Eight global clock buffers Programmable output slew rate Internal 3-state busses RAM for FIFOs and registers JTAG for board test Fast carry logic for arithmetic Wide decode 12 mA output drive, 24 mA per pair (24 mA/48 mA for A/H families)
Process	CMOS EPROM	CMOS EPROM	CMOS static RAM	CMOS static RAM
Programming Method	PROM programmer OTP or UV erasable Configuration on chip	PROM programmer OTP or UV erasable Configuration on chip	Programmed in circuit Four modes Configuration stored externally	Programmed in circuit Six modes Configuration stored externally
Re-programmable	Yes – after UV erasure	Yes – after UV erasure	Yes – in milliseconds Reprogrammable in circuit	Yes – in milliseconds Reprogrammable in circuit
Factory Tested	Yes	Yes	Yes	Yes
Key Applications	Complex state machines Complex counters Bus & peripheral interface Memory control PAL-cruncher Accumulators/ incrementors Magnitude/window comparators	High speed graphics Multiport memory controllers High speed bus interface 51 MHz, 18 bit accumulators DMA controllers Wide decoders High speed state machines Complex controllers	Simple state machines General logic replacement Reprogrammable applications Battery-powered logic 3 V operation Very fast counters	Simple state machines Complex logic replacement Board integration Adders/comparators Reprogrammable applications RAM application: FIFOs, buffers Fast/compact counters JTAG boards Bus interfacing

X5233

Speed and Density

	EPLDs			FPGAs		
	XC7200A (-16)	XC7336 (-7)*	XC7354 (-10)	XC3000A (-7)	XC3100 (-3)	XC4000 (-5)
16-Bit Synchronous Binary Counter	60 MHz	91 MHz	83 MHz	51 MHz 24 CLBs	102 MHz 24 CLBs	111 MHz 17 CLBs
16-Bit Unidirectional Loadable Counter **Max Density**	60 MHz	91 MHz	83 MHz	18 MHz 16 CLBs	31 MHz 16 CLBs	43 MHz 9 CLBs
Max Speed	60 MHz	91 MHz	83 MHz	32 MHz 24 CLBs	55 MHz 24 CLBs	43 MHz 9 CLBs
16-Bit U/D Counter **Max Density**	60 MHz	91 MHz	83 MHz	15 MHz 16 CLBs	28 MHz 16 CLBs	43 MHz 9 CLBs
Max Speed	60 MHz	91 MHz	83 MHz	30 MHz 27 CLBs	50 MHz 27 CLBs	43 MHz 9 CLBs
8:1 Multiplexer	22 ns	7.5 ns	10 ns	16 ns 8 CLBs	10 ns 8 CLBs	16 ns 5 CLBs
12-Bit Decode From Input Pad	22 ns	7.5 ns	10 ns	15 ns 4 CLBs	12 ns 4 CLBs	12 ns 0 CLBs
16-Bit Accumulator	42 MHz	n/a	51 MHz	21 MHz 29 CLBs	36 MHz 29 CLBs	39 MHz 9 CLBs
Data Path [1]	60 MHz	91 MHz	83 MHz	50 MHz 16 CLBs	95 MHz 16 CLBs	85 MHz 12 CLBs
Timer Counter [2]	60 MHz	91 MHz	83 MHz	28 MHz 23 CLBs	52 MHz 23 CLBs	40 MHz 12 CLBs
State Machine [3]	60 MHz	91 MHz	83 MHz	18 MHz 34 CLBs	30 MHz 34 CLBs	30 MHz 26 CLBs
Arithmetic [4]	22 MHz	n/a	34 MHz	17 MHz 23 CLBs	29 MHz 23 CLBs	20 MHz 16 CLBs
16 Channel, 32-Bit DMA	n/a	n/a	n/a	n/a n/a	n/a n/a	20 MHz 72 CLBs

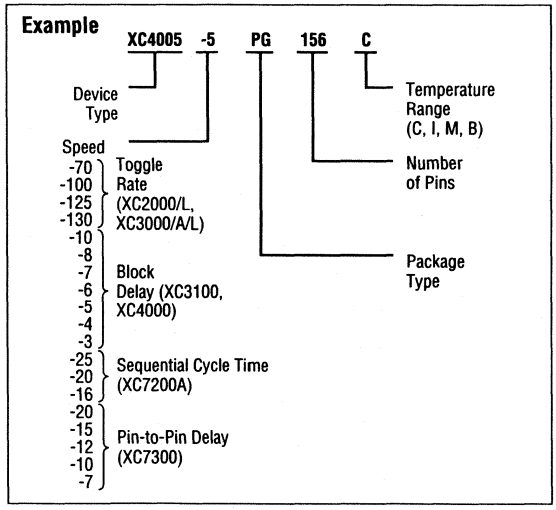
* Estimated

Notes:
1. 32 inputs, 4:1 mux, register, 8-bit shift register
2. 8-bit T/C, latch, mux, compare
3. 16 states, 40 transistions, 10 inputs, 8 outputs
4. 4x4 multiplier, 8-bit accumulator

All speeds are worst case temperature and voltage.

X5230

Ordering Information



X5231

Product Comparison

		Typical Gates	Typical Gates Using RAM	Max I/Os	Flip-Flops	RAM bits	Macrocells	Available CLBs
EPLDs	**XC7200A Family**							
	XC7236A	1600	n/a	36	68	n/a	36	n/a
	XC7272A	3200	n/a	72	120	n/a	72	n/a
	XC7300 Family							
	XC7336	1000	n/a	38	36	n/a	36	n/a
	XC7354	2200	n/a	66	108	n/a	54	n/a
	XC7372	3000	n/a	84	126	n/a	72	n/a
	XC73108	4600	n/a	120	198	n/a	108	n/a
	XC73144	6200	n/a	156	234	n/a	144	n/a
FPGAs	**XC2000 Family**							
	XC2064/L	0.6k–1.0k	n/a	58	122	0	n/a	64
	XC2018/L	1.0k–1.5k	n/a	74	174	0	n/a	100
	XC3000/XC3100 Family							
	XC3020/A/L/XC3120	1.3k–1.8k	n/a	64	256	0	n/a	64
	XC3030/A/L/XC3130	2.0k–2.7k	n/a	80	360	0	n/a	100
	XC3042/A/L/XC3142	3.0k–3.7k	n/a	96	480	0	n/a	144
	XC3064/A/L/XC3164	4.0k–5.5k	n/a	120	688	0	n/a	224
	XC3090/A/L/XC3190	5.0k–7.5k	n/a	144	928	0	n/a	320
	XC3195	6.5k–9.0k	n/a	176	1,320	0	n/a	484
	XC4000 Family							
	XC4002A	1.6k–2.0k	2.2k–2.8k*	64	256	2,048	n/a	64
	XC4003/XC4003A	2.5k–3.0k	3.5k–4.2k*	80	360	3,200	n/a	100
	XC4003H	2.5k–3.0k	3.5k–4.2k*	160	200	3,200	n/a	100
	XC4004A	3.2k–4.0k	4.6k–5.6k*	96	480	4,608	n/a	144
	XC4005/XC4005A	4.0k–5.0k	6.0k–7.0k*	112	616	6,272	n/a	196
	XC4005H	4.0k–5.0k	6.0k–7.0k*	192	392	6,272	n/a	196
	XC4006	5.0k–6.0k	7.5k–8.5k*	128	768	8,192	n/a	256
	XC4008	6.5k–8.0k	9.7k–11.2k*	144	936	10,368	n/a	324
	XC4010	8.0k–10.0k	12.0k–14.0k*	160	1,120	12,800	n/a	400
	XC4010D	8.0k–10.0k	n/a	160	1,120	12,800	n/a	400
	XC4013	10.0k–13.0k	14.0k–16.0k*	192	1,536	18,432	n/a	576
	XC4025	20.0k–25.0k	30.0k–35.0k*	256	2,560	32,768	n/a	1,024

* Assumes 10% of device used as RAM.

X5232

Package Options

	Surface Mount						Through-hole
	PLCC	PQFP	TQFP	VQFP	CQFP	BGA	PGA
Standard	JEDEC	EIAJ	EIAJ	EIAJ	JEDEC	JEDEC	JEDEC
Lead Pitch	50 mil	0.65/0.5 mm	0.5 mm	0.5 mm	25 mil	1.5 mm	100 mil
Body	Plastic	Plastic	Plastic	Plastic	Ceramic	FR4	Ceramic/Plastic
Temp Options	C, I	C, I	C, I	C, I	M, B	C	C, I, M, B
Ordering Code	PC	PQ	TQ	VQ	CB	WB, PB	PG, PP
XC7236A	44						
XC7272A	68, 84						84
XC7336	44		100	100			
XC7354	44, 68						
XC7372	68, 84						
XC3108	84	160				225	144
XC73144						225	184
XC2064	44, 68						68
XC2018	44, 68, 84		100	100			84
XC3020/XC3120	68, 84	100			100		84
XC3030/XC3130	44, 68, 84	100	100	100			84
XC3042/XC3142	84	100	100	100	100		84, 132
XC3064/XC3164	84	160					132
XC3090/XC3190	84	160, 208			164		175
XC3195	84	160, 208					175, 223
XC4002A	84	100	100		100		120
XC4003/XC4003A	84	100	100				120
XC4003H		208					191
XC4004A	84	160	144				120
XC4005/XC4005A	84	160, 208	144		164		156
XC4005H		240					223
XC4006		160, 208					156
XC4008		208			196		191
XC4010/XC4010D	84	160, 208			196	225	191
XC4013		208, 240					223
XC4025		208				225	299

X3469

Number of Available I/O Pins

	Max	Number of Package Pins																			
	I/O	44	64	68	84	100	120	132	144	156	160	164	175	176	184	191	196	208	223	225	240	299
XC7236A	36	36																				
XC7272A	72			56	72																	
XC7336	38	38																				
XC7354	58	38		58																		
XC7372	72			56	72																	
XC73108	120				72				120		120		120		120					120		
XC73144	156														156					156		
XC2064	58	34		58																		
XC2018	74	34		64	74	74																
XC3020/XC3120	64		54	58	64	64																
XC3030/XC3130	80	34	54	58	74	80																
XC3042/XC3142	96				74	82		96														
XC3064/XC3164	120				70			110			120											
XC3090/XC3190	144				70						138	142	144	144				144				
XC3195	176				70						138		144					176	176			
XC4002A	64				61	64	64															
XC4003/XC4003A	80				61	77	80															
XC4003H	160															160		160				
XC4004A	96				61		95		96		96											
XC4005/XC4005A	112				61				112	112	112	112						112				
XC4005H	192																		192		192	
XC4006	128									125	128							128				
XC4008	144															144	144	144				
XC4010/XC4010D	160				61											160	160	160		160		
XC4013	192																	160	192	192	192	
XC4025	256																		192		193	256

X3406

HardWire Gate Arrays

The No-Risk Gate-Array Migration Path For Xilinx FPGAs

The HardWire gate array provides an easy, transparent migration path – providing a low-cost, no-risk solution for high-volume-production applications.

Unlike ordinary, general-purpose gate arrays, the HardWire gate array is architecturally identical to its FPGA counterpart. The programmable elements in the FPGA are simply removed and replaced with fixed metal connections. The resulting HardWire gate array die is considerably smaller and lower cost.

Convert With Confidence

The HardWire gate array offers the same proven, qualified process technology as the FPGA it replaces. And, since the architectures are identical, FPGAs and HardWire gate arrays have similar timing.

In addition, the interchangeability of the FPGA and the HardWire gate array means that FPGAs can always be substituted – to quickly boost production to meet demand, or to avoid gate-array inventory worries toward the end of the product life cycle.

The Fastest, Easiest Way To Save

Converting from an FPGA to a HardWire gate array couldn't be easier. The mask and test programs are generated by Xilinx from the user's existing FPGA file. The time-consuming and costly re-design and resimulation usually associated with FPGA-to-gate array migration is virtually eliminated, along with the risk.

Xilinx built-in test logic and Automatic Test Generation (ATG) software guarantee 100% fault coverage, while eliminating the need for test vectors. With migration this simple, designers spend less time on rework and more time on new projects.

In addition to engineering savings from easy conversion and the elimination of opportunity costs, the HardWire gate array architecture also means that NRE costs are low – usually <$10K (depending on size). The HardWire gate array offers typical device savings of 50 - 80% over the equivalent FPGA.

For more information and to request the HardWire Data Book, contact the nearest Xilinx Sales Office.

Military Devices

Xilinx was the first company to offer military FPGAs by introducing 883 qualified versions of the XC2000 and XC3000 Families in 1989. The MIL-STD-883 qualified versions of our XC4000 Family will soon be available. These products offer a number of key benefits to military users.

Increased Design Flexibility. Xilinx parts are standard production ASICs, where one spec can be used for multiple applications. Since there is no fab turnaround time, design changes can be made in minutes, reducing product development time. In addition, our Class B devices are available from distributor stock.

Reprogrammability. Because Xilinx parts are reprogrammable, design changes can be made while in production. And, the same logic can be used for multiple, nonconcurrent tasks.

Low Total Cost. Because there are no non-recurring engineering (NRE) costs, Xilinx devices are very cost effective for military volumes.

Reliable. Our parts are fully compliant to MIL-STD-883 Class B, with very low FIT rates. Products built and tested to Standard Military Drawings (SMDs) are also available.

Fully Tested. Because our parts are fully tested with 100% fault coverage, the user need not generate test programs or vectors.

Available in die form. Xilinx is the only vendor supplying FPGAs in die form, tested and qualified for use in military hybrids and multi-chip modules. These are available through Chip Supply, Melbourne, Florida, at (407) 298-7100.

For more information on military devices, contact the nearest Xilinx Sales Office.

A Guide to Xilinx Part Numbers

FPGA, LCA, EPLD, CPLD, PLD

FPGA (Field Programmable Gate Array) is the generic name for all channel-routed, user-programmable logic devices like the XC2000, XC3000, and XC4000 families. LCA (Logic Cell Array), a reference to the Xilinx FPGA architecture, is a Xilinx trademark, and should, therefore, only be used as an adjective e.g., LCA device, LCA architecture. EPLDs (Erasable Programmable Logic Devices or EPROM-based Programmable Logic Devices) are more structured, coarse-grained devices based on a PAL architecture (PAL is an AMD trademark), like the XC7200 and XC7300 families. CPLD (Complex Programmable Logic Device) typically is used as a more generic-sounding synonym for EPLD. PLD (Programmable Logic Device) describes all programmable logic, from the smallest PAL device to the largest FPGA. To confuse matters, the trade press and some vendors occasionally use FPGA and/or CPLD as an all-encompassing label for both channel-routed and PAL-structured high-density PLDs.

XC2064 to XC4025

The XC2064 was the first Xilinx device and the first FPGA. There never was an XC1000 family. XC became the recognized Xilinx prefix, like i for Intel or Am for AMD.

The very first device was called the XC2064 to describe its 8 x 8 = 64 CLB structure. This nomenclature was never used again; all later Xilinx FPGAs use the two trailing digits to describe the claimed gate density, expressed in multiples of 100 gates for the XC2000 and XC3000 families, and in multiples of 1000 gates for the XC4000 family.

The XC2000 and XC3000 families count the highest number of gates that can be implemented, assuming that all function generators and flip-flops are being used. A typical gate-count value is 40% lower, since some function generators may be used to implement simple logic, some flip-flops may be unused, and some CLBs may not be routable. The XC4000 family uses a more conservative method of counting gates. A typical design will pack the claimed number of gates, and substantially more when some function generators are used as RAM.

XC3100 describes a speed-enhanced version of the XC3000 family, with identical functionality and bitstream compatibility. The XC3195 is a new member with 50% higher CLB count than the XC3090 or XC3190.

XC2300, XC3300, and XC4300 are non-programmable HardWire derivatives of the respective FPGA families, offering lower cost for high-volume production.

XC7000 describes the Xilinx EPLD offerings. The original devices are the XC7236 and XC7272, with 36 and 72 Macrocells respectively. The new devices using the Dual Block Architecture are called the XC7336 through XC73108, with 36 to 108 Macrocells.

What about the A?

The different product lines use the suffix A for different purposes.

XC3000A, XC3100A

XC3000A and XC3100A are new, functionally enhanced variations of the XC3000 and XC3100 families.They have better connectivity, bitstream error checking, and better start-up behavior. Any XC3000 or XC3100 design can be upgraded to the XC3000A or XC3100A with no change, not even to the bitstream.

XC4000A

XC4000A describes the smaller members of the XC4000 family, that do not need the generous interconnect structure of the larger devices. Only the XC4003 and XC40005 exist in both flavors: the XC4003A and XC4005A have the optimized interconnect structure, and are less expensive; the XC4003 and XC4005 have the full interconnect structure, and are more expensive. Though not bitstream compatible, A and non-A devices come in the same packages and with the same pinout.

XC7236A and XC7272A

The "A" parts are functionally identical to the original parts, but are manufactured on a more advanced process, requiring a different programming algorithm with a lower programming voltage. This mandates a new part number, although the "A" parts behave identically in the user application.

XC1736A

The XC1736 was the original device. When it was changed to a more advanced process, the programming algorithm and voltage changed. This mandates a new part number, although the "A" part behaves identically in the user application.

The new, redesigned and unified family of XC17000 devices uses the suffix D, since A was already used for the XC1736A, B denotes military devices, and C is overused to indicate commercial temperature or ceramic packages.

XC4010D

This is a low-cost version of the XC4010 without on-chip RAM.

Speed Designations

Early Xilinx FPGA devices, XC2000 through XC3000, describe device speed by the guaranteed worst-case toggle rate of the CLB flip-flops. As devices become faster, this toggle frequency approaches 300 MHz, and is no longer a meaningful indicator. The newer families, starting with the XC4000 and XC3100, use the combinatorial delay through a function generator, T_{ILO}, expressed in nanoseconds, to designate speed.

EPLDs designate device speed by the fastest combinatorial delay from input pin to output pin, in nanoseconds.

There is no speed designation in the Serial PROM part number; see the data sheet for details.

XILINX®

FPGA Product Description and Specifications

A Technical Overview For the First-Time User

In the XC2000, XC3000, and XC4000 devices, Xilinx offers three evolutionary and compatible generations of Field Programmable Gate Arrays (FPGAs). Here is a short description of their common features.

Every Xilinx FPGA performs the function of a custom LSI circuit, like a gate array, but the Xilinx device is user-programmable and even reprogrammable in the system. Xilinx sells standard off-the-shelf devices in three families, and many different sizes, speeds, operating temperature ranges, and packages. The user selects the appropriate Xilinx device, and then converts the design idea or schematic into a configuration data file, using the Xilinx development system software running on a PC or workstation, and loads this file into the Xilinx FPGA.

This overview describes two different aspects of the Xilinx FPGA,

- what kind of user-defined logic it can implement, and
- how the device is programmed.

User Logic

Different in structure from traditional logic circuits, PALs, EPLDs and even gate arrays, all Xilinx FPGAs implement combinatorial logic in small look-up tables (16 x 1 ROMs); each such table either feeds the D-input of a flip-flop or drives other logic or I/O. Each device contains a matrix of identical logic blocks, usually square, from 8 x 8 in the XC2064 to 32 x 32 in the XC4025. Short and long metal lines run horizontally and vertically in-between these logic blocks, selectively interconnecting them or connecting them to the input/output blocks.

This modular architecture is rich in registers and powerful function generators that can implement any function of up-to-five variables, all with the same speed. For wider inputs, function generators are easily concatenated. Generous on-chip buffering makes block delays insensitive to loading by the interconnect structure, but all interconnect delays are layout-dependent and must be analyzed if the design is performance-critical.

Clock lines are well-buffered and can drive all flip-flops with < 2 ns skew from corner to corner, even throughout the biggest device. The user need not worry about clock loading or clock-delay balancing, or about hold-time issues on the chip, if the designated clock lines (eight in the XC4000 devices, two in all other devices) are used.

XC3000/3100 and XC4000 devices can implement internal bidirectional busses. The XC4000 devices have dedicated fast carry circuits that improve the efficiency and speed of adders, subtractors, comparators, accumulators and synchronous counters. XC4000 also supports boundary scan on each pin.

Almost all device pins are available as bidirectional user I/O, with the exception of 4 to 24 supply connections (V_{CC} and GND) and a few pins dedicated to the configuration process. All inputs and outputs within each family have identical electrical characteristics, but output current capability varies among families: The XC2000 and XC3000 outputs can sink and source 4 mA, XC3100 can sink 8 mA, XC4000 12 mA and XC4000H 24 mA. The outputs on XC2000/XC3000/XC3100 devices swing rail-to-rail, while XC4000 outputs are n-channel-only, "totem-pole", with lower V_{OH} for higher speed.

XC2000/XC3000/XC3100 inputs can be globally programmed for either TTL-like input thresholds or CMOS thresholds. XC4000 has fixed TTL-like input thresholds. All inputs have hysteresis (Schmitt-trigger action) of 100 to 200 mV.

All Xilinx FPGAs have a global asynchronous reset input affecting all device flip-flops. In the XC4000-family devices, any pin can be configured as a reset input, in the other families, RESET is a dedicated pin.

Since all Xilinx FPGAs use CMOS-SRAM technology, their quiescent or stand-by power consumption is very low, a few microwatts for XC2000/XC3000 devices and max 25 mW for XC3100 , max 50 mW for XC4000 devices. The operational power consumption is totally dynamic, proportional to the rate of change of inputs, outputs, and internal nodes. Typical power consumption is between 100 mW and 2 W, depending on the device size.

XC2000 and XC3000 devices can be powered-down and their configuration can be maintained by a >2.3 V battery. Current consumption is only a few microamps. The device 3-states all outputs, ignores all inputs, and resets its flip-flops, but retains its configuration.

XC3000/XC3100/XC4000 devices monitor V_{CC} continuously and shut down when they detect a V_{CC} drop to 3 V. The device then 3-states all outputs and prepares for reconfiguration. XC2000 devices need an external monitor, if there is any danger of V_{CC} dropping significantly without going all the way to ground.

Continued on Page 2-4

Component Availability

Pins	44	48	64	68		84		100				120	132		144
Type	Plastic PLCC	Plastic DIP	Plastic VQFP	Plastic PLCC	Ceramic PGA	Plastic PLCC	Ceramic PGA	Plastic PQFP	Plastic TQFP	Plastic VQFP	Top-Brazed CQFP	Ceramic PGA	Plastic PGA	Ceramic PGA	Plastic TQFP
Code	PC44	PD48	VQ64	PC68	PG68	PC84	PG84	PQ100	TQ100	VQ100	CB100	PG120	PP132	PG132	TQ144
XC2064	✔	✔		✔	✔										
XC2018			✔	✔		✔	✔		✔						
XC2064L			✔	✔											
XC2018L			✔			✔				✔					
XC3020				✔		✔	✔	✔			✔				
XC3030	✔			✔		✔	✔	✔	✔						
XC3042						✔	✔	✔	✔		✔				
XC3064						✔							✔	✔	
XC3090						✔							✔	✔	
XC3020A				✔		✔	✔	✔							
XC3030A	✔		✔	✔		✔	✔	✔		✔					
XC3042A						✔	✔	✔		✔					✔
XC3064A						✔							✔	✔	✔
XC3090A						✔							✔	✔	
XC3020L			✔			✔									
XC3030L			✔			✔				✔					
XC3042L						✔				✔					✔
XC3064L						✔									✔
XC3090L						✔									
XC3120				✔		✔	✔	✔			✔				
XC3130	✔			✔		✔	✔	✔	✔						
XC3142						✔	✔	✔	✔		✔		✔	✔	✔
XC3164						✔							✔	✔	
XC3190						✔									
XC3195						✔									
XC4002A						✔		✔		✔		✔			
XC4003						✔						✔			
XC4003A						✔		✔		✔	✔	✔			
XC4003H															
XC4004A						✔						✔			✔
XC4005						✔									
XC4005A						✔									✔
XC4005H															
XC4006						✔									
XC4008						✔									
XC4010						✔									
XC4010D						✔									
XC4013															
XC4025															

X3455

Component Availability (continued)

Pins	156	160	164	175		176	191	196	208		223	225	240		299
Type	Ceramic PGA	Plastic PQFP	Top-Brazed CQFP	Plastic PGA	Ceramic PGA	Plastic TQFP	Ceramic PGA	Top-Brazed CQFP	Plastic PQFP	Metal MQFP	Ceramic PGA	Ceramic BGA	Plastic PQFP	Metal MQFP	Ceramic PGA
Code	PG156	PQ160	CB164	PP175	PG175	TQ176	PG191	CB196	PQ208	MQ208	PG223	CG225	PQ240	MQ240	PG299
XC2064															
XC2018															
XC2064L															
XC2018L															
XC3020															
XC3030															
XC3042															
XC3064		✔													
XC3090		✔	✔	✔	✔				✔						
XC3020A															
XC3030A															
XC3042A															
XC3064A		✔													
XC3090A		✔		✔	✔	✔			✔						
XC3020L															
XC3030L															
XC3042L															
XC3064L															
XC3090L						✔									
XC3120															
XC3130															
XC3142															
XC3164		✔													
XC3190		✔	✔	✔	✔				✔						
XC3195		✔		✔	✔				✔		✔				
XC4002A															
XC4003															
XC4003A															
XC4003H							✔		✔						
XC4004A		✔													
XC4005	✔	✔	✔						✔						
XC4005A	✔	✔							✔						
XC4005H											✔		✔	✔	
XC4006	✔	✔							✔						
XC4008		✔					✔		✔	✔					
XC4010		✔					✔	✔	✔	✔		✔			
XC4010D		✔													
XC4013										✔	✔	✔	✔	✔	
XC4025										✔					✔

X3456

Programming or Configuring the Device

A design usually starts as a block diagram or schematic, drawn with one of the popular CAE tools, e.g. ViewDraw. Many of these tools have an interface to XACT, the Xilinx development system, running on PCs or popular workstations.

After schematic- or equation-based entry, the design is automatically converted to a Xilinx Netlist Format (XNF). The XACT software first partitions the design into logic blocks, then finds a near-optimal placement for each block, and finally selects the interconnect routing. This process of Partitioning, Placement, and Routing (PPR) runs automatically, but the user may also affect the outcome by imposing specific constraints, or selectively editing critical portions of the design, using the graphic Design Editor (XDE). The user thus has a wide range of choices between a fully automatic implementation and detailed involvement in the layout process.

Once the design is complete, it is documented in an LCA file, from which a serial bitstream file can be generated.

The user then exercises one of several options to load this file into the Xilinx FPGA device, where it is stored in latches, arranged to resemble one long shift register. The data content of these latches customizes the FPGA to perform the intended digital function. The number of configuration bits varies with device type, from 12,038 bits for the smallest device (XC2064) to 422,168 bits for the largest device presently available (XC4025). Multiple LCA devices can be daisy-chained and configured with a common, concatenated bitstream. Device utilization does not change the number of configuration bits.

Inside the device, these configuration bits control or define the combinatorial circuitry, flip-flops, interconnect structure, and the I/O buffers. Upon power-up, the device waits for V_{CC} to reach an acceptable level, then clears the configuration memory, holds all internal flip-flops reset, and 3-states almost all outputs but activates their weak pull-up resistors. The device then initiates configuration, either as a master, clocking a serial PROM to receive the serial bitstream, or as a slave, accepting an external clock and serial or 8-bit parallel data from an external source.

The Xilinx serial PROM is the simplest way to configure the device, using only four device pins. Typical configuration time is around 1 µs per bit, but there are ways to reduce it by a factor of up to ten. Configuration thus takes from a few to a few hundred milliseconds. Xilinx serial PROMs come in sizes from 18,000 to 128K bits (256K bits in the near future); PROMs can also be daisy-chained to store a longer bitstream.

The LCA device can also be configured with byte-wide data, either from an industry-standard PROM or from a microprocessor. The LCA device drives the PROM addresses directly, or it handshakes with the microprocessor like a typical peripheral. The byte-wide data is immediately converted into an internal serial bitstream, clocked by the internal Configuration Clock (CCLK). Parallel configuration modes are, therefore, not faster than serial modes.

The user can reconfigure the device at any time by pulling the DONE pin Low, which instigates a new configuration sequence. During this process, all outputs not used for configuration are 3-stated. Partial re-configuration is not possible.

After the device has been programmed, the content of the configuration "shift register" can be read back serially, without interfering with device operation. XC4000 devices include a synchronized simultaneous transfer of all user-register information into the configuration registers. This adds in-circuit-emulation capability to the readback function.

XC4000 Logic Cell Array Families

Table of Contents

Overview

Introduced in 1990, the XC4000 family has found rapid acceptance by demanding users. The RAM capability offers a new freedom to design, the dedicated carry logic speeds up arithmetic and counters, and the wide decoders eliminate the need for external decoding.

Xilinx has met this enthusiastic user response with the rapid introduction of new device types. Stretching from 2,000 to 25,000 gate capacity, the XC4000 family now has 14 part types available.

The XC4005, XC4006, XC4008, XC4010, XC4013 and XC4025 represent the original concept, a structure with abundant routing resources to accomodate even the most complex design. Since smaller devices require disproportionally less routing resources, the low-end XC4000A family saves silicon area and thus cost by having fewer interconnect lines. The XC4000A family consists of the XC4002A, XC4003A, XC4004A, and XC4005A. At the XC4005 level, both device types are available; the XC4005A is more economical, the XC4005 has more routing resources. Since the devices are pin-compatible, the user can start the design with the sure-to-route XC4005, then later switch to the more economical XC4005A, if it is sufficient to implement the design.

Some applications require more I/O than available in the XC4000 and XC4000A families. This is especially true in very large logic emulators where many XC4000 devices are interconnected in a big matrix of devices. In these applications, the classical Xilinx FPGA structure with two IOBs at each end of each CLB row and column represents an I/O bottleneck. For these and similar applications, Xilinx offers the XC4003H and XC4005H devices with approximately twice the I/O count of the corresponding XC4000 device.

Xilinx Now Guarantees Pin-to-Pin Timing

Input set-up and hold times as well as output delays are traditionally specified with respect to the internal clock. That makes these parameters independent of device size, since it is only the clock delay that differs between different size devices of one family and one speed designation.

To arrive at the real pin-to-pin values, the user has to include the on-chip clock delay by adding it to the output delay, subtracting it from the input set-up time, and adding it to the hold time value.

The pin-to-pin set-up time can be difficult to assess, because it is not obvious what percentage of the clock delay should be subtracted. Subtracting 100% of the worst-case value would be overly optimistic, assuming perfect delay tracking on the chip. Xilinx recommends that the user subtract 70% of the guaranteed maximum clock delay. This may be overly conservative, but is a safe method that understates device performance.

For XC4000 devices, the Data Book now publishes tested and guaranteed pin-to-pin parameters. These published values are slightly better than the values calculated from the speeds files. The user can be assured that these 100% tested pin-to-pin values are more authentic than the calculated values. This is the only case where the Data Book numbers have a higher authority than the data in the speeds files and other simulation and report files.

Here are two examples, using the XC4005-5:

Clock-to-Output Delay:

calculated by adding worst-case T_{OKPOF} (clock-to-pad delay, fast slew rate) and T_{PG} (global clock delay):

7.0 ns + 6.0 ns = 13.0 ns

measured and guaranteed:
13.0 ns (i.e., same as calculated)

Input Set-up Time:

calculated by subtracting 70% of T_{PG} (global clock delay) from T_{PICKD} (set-up time, pad-to-clock [IK] with delay):

24.0 ns - (.70 * 6.0 ns) = 19.8 ns

measured and guaranteed:
18.0 ns (i.e., 1.8 ns better than calculated)

XC4000, XC4000A, XC4000H Logic Cell Array Families

Product Description

Features

- Third Generation Field-Programmable Gate Arrays
  - Abundant flip-flops
  - Flexible function generators
  - On-chip ultra-fast RAM
  - Dedicated high-speed carry-propagation circuit
  - Wide edge decoders
  - Hierarchy of interconnect lines
  - Internal 3-state bus capability
  - Eight global low-skew clock or signal distribution network
- Flexible Array Architecture
  - Programmable logic blocks and I/O blocks
  - Programmable interconnects and wide decoders
- Sub-micron CMOS Process
  - High-speed logic and Interconnect
  - Low power consumption
- Systems-Oriented Features
  - IEEE 1149.1-compatible boundary-scan logic support
  - Programmable output slew rate
  - Programmable input pull-up or pull-down resistors
  - 12-mA sink current per output (XC4000 family)
  - 24-mA sink current per output (XC4000A and XC4000H families)
- Configured by Loading Binary File
  - Unlimited reprogrammability
  - Six programming modes
- XACT Development System runs on '386/'486-type PC, NEC PC, Apollo, Sun-4, and Hewlett-Packard 700 series
  - Interfaces to popular design environments like Viewlogic, Mentor Graphics and OrCAD
  - Fully automatic partitioning, placement and routing
  - Interactive design editor for design optimization
  - 288 macros, 34 hard macros, RAM/ROM compiler

Description

The XC4000 families of Field-Programmable Gate Arrays (FPGAs) provide the benefits of custom CMOS VLSI, while avoiding the initial cost, time delay, and inherent risk of a conventional masked gate array.

The XC4000 families provide a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), interconnected by a powerful hierarchy of versatile routing resources, and surrounded by a perimeter of programmable Input/Output Blocks (IOBs).

XC4000-family devices have generous routing resources to accommodate the most complex interconnect patterns. XC4000A devices have reduced sets of routing resources, sufficient for their smaller size. XC4000H high I/O devices maintain the same routing resources and CLB structure as the XC4000 family, while nearly doubling the available I/O.

The devices are customized by loading configuration data into the internal memory cells. The FPGA can either actively read its configuration data out of external serial or byte-parallel PROM (master modes), or the configuration data can be written into the FPGA (slave and peripheral modes).

The XC4000 families are supported by powerful and sophisticated software, covering every aspect of design: from schematic entry, to simulation, to automatic block placement and routing of interconnects, and finally the creation of the configuration bit stream.

Since Xilinx FPGAs can be reprogrammed an unlimited number of times, they can be used in innovative designs where hardware is changed dynamically, or where hardware must be adapted to different user applications. FPGAs are ideal for shortening the design and development cycle, but they also offer a cost-effective solution for production rates well beyond 1000 systems per month.

Table 1. The XC4000 Families of Field-Programmable Gate Arrays

Device	XC4002A	4003/3A	4003H	4004A	4005/5A	4005H	4006	4008	4010/10D	4013	4025
Appr. Gate Count	2,000	3,000	3,000	4,000	5,000	5,000	6,000	8,000	10,000	13,000	25,000
CLB Matrix	8 x 8	10 x 10	10 x 10	12 x 12	14 x 14	14 x 14	16 x 16	18 x 18	20 x 20	24 x 24	32 x 32
Number of CLBs	64	100	100	144	196	196	256	324	400	576	1,024
Number of Flip-Flops	256	360	200	480	616	392	768	936	1,120	1,536	2,560
Max Decode Inputs (per side)	24	30	30	36	42	42	48	54	60	72	96
Max RAM Bits	2,048	3,200	3,200	4,608	6,272	6,272	8,192	10,368	12,800*	18,432	32,768
Number of IOBs	64	80	160	96	112	192	128	144	160	192	256

*XC4010D has no RAM

XC4000 Compared to XC3000

For those readers already familiar with the XC3000 family of Xilinx Field Programmable Gate Arrays, here is a concise list of the major new features in the XC4000 family.

CLB has two **independent** 4-input function generators.
A **third** function generator combines the outputs of the two other function generators with a ninth input.
All function inputs are swappable, all have full access; none are mutually exclusive.
CLB has **very fast arithmetic carry** capability.
CLB function generator look-up table can also be used as high-speed **RAM.**
CLB flip-flops have asynchronous set **or** reset.
CLB has **four outputs**, two flip-flops, two combinatorial.
CLB connections symmetrically located on all **four** edges.

IOB has more versatile clocking polarity options.
IOB has programmable input set-up time:
long to avoid potential hold time problems,
short to improve performance.
IOB has Longline access through its own TBUF.
Outputs are **n-channel only**, lower V_{OH} increases speed.
XC4000 outputs can be paired to double sink current to **24 mA.** XC4000A and XC400H outputs can each sink 24 mA, can be paired for **48 mA** sink current.

IEEE 1149.1-type **boundary scan** is supported in the I/O.

Wide decoders on all four edges of the LCA device.

Increased **number of interconnect resources.**
All CLB inputs and outputs have **access to most interconnect lines.**
Switch Matrices are simplified to increase speed.
Eight global nets can be used for clocking or distributing logic signals.
TBUF output configuration is more versatile and 3-state control less confined.

Program is single-function input pin,overrides everything.
$\overline{\text{INIT}}$ pin also acts as Configuration Error output.

Peripheral Synchronous Mode (8 bit) has been added.
Peripheral Asynchronous Mode has improved handshake.
Start-up can be **synchronized** to any user clock (this is a configuration option).
No Powerdown, but instead a **Global 3-state input** that does not reset any flip-flops.
No on-chip **crystal oscillator** amplifier.

Configuration Bit Stream includes **CRC error checking.**
Configuration Clock can be increased to **>8 MHz.**
Configuration Clock is **fully static**, no constraint on the maximum Low time.

Readback either ignores flip-flop content (avoids need for masking) or it takes a **snapshot** of all flip-flops at the start of Readback.
Readback has same **polarity** as Configuration and can be **aborted.**

Table 2. Three Generations of Xilinx Field-Programmable Gate Array Families

Parameter	XC4013	XC3090/3190	XC2018
Number of flip-flops	1,536	928	174
Max number of user I/O	192	144	74
Max number of RAM bits	18,432	0	0
Function generators per CLB	3	2	2
Number of logic inputs per CLB	9	5	4
Number of logic outputs per CLB	4	2	2
Number of low-skew global nets	8	2	2
Dedicated decoders	yes	no	no
Fast carry logic	yes	no	no
Internal 3-state drivers	yes	yes	no
Output slew-rate control	yes	yes	no
Power-down option	no	yes	yes
Crystal oscillator circuit	no	yes	yes

Architectural Overview

The XC4000 families achieve high speed through advanced semiconductor technology and through improved architecture, and supports system clock rates of up to 50 MHz. Compared to older Xilinx FPGA families, the XC4000 families are more powerful, offering on-chip RAM and wide-input decoders. They are more versatile in their applications, and design cycles are faster due to a combination of increased routing resources and more sophisticated software. And last, but not least, they more than double the available complexity, up to the 20,000-gate level.

The XC4000 families have 13 members, ranging in complexity from 2,000 to 25,000 gates.

Logic Cell Array Families

Xilinx high-density user-programmable gate arrays include three major configurable elements: configurable logic blocks (CLBs), input/output blocks (IOBs), and interconnections. The CLBs provide the functional elements for constructing the user's logic. The IOBs provide the interface between the package pins and internal signal lines. The programmable interconnect resources provide routing paths to connect the inputs and outputs of the CLBs and IOBs onto the appropriate networks. Customized configuration is established by programming internal static memory cells that determine the logic functions and interconnections implemented in the LCA device.

The first generation of LCA devices, the XC2000 family, was introduced in 1985. It featured logic blocks consisting of a combinatorial function generator capable of implementing 4-input Boolean functions and a single storage element. The XC2000 family has two members ranging in complexity from 800 to 1500 gates.

In the second-generation XC3000 LCA devices, introduced in 1987, the logic block was expanded to implement wider Boolean functions and to incorporate a second flip-flop in each logic block. Today, the XC3000 devices range in complexity from 1,300 to 10,000 usable gates. They have a maximum guaranteed toggle frequency ranging from 70 to 270 MHz, equivalent to maximum system clock frequencies of up to 80 MHz.

The third generation of LCA devices further extends this architecture with a yet more powerful and flexible logic block. I/O block functions and interconnection options have also been enhanced with each successive generation, further extending the range of applications that can be implemented with an LCA device.

This third-generation architecture forms the basis of the XC4000 families of devices that feature logic densities up to 25,000 usable gates and support system clock rates of up to 50 MHz. The use of an advanced, sub-micron CMOS process technology as well as architectural improvements contribute to this increase in FPGA capabilities. However, achieving these high logic-density and performance levels also requires new and more powerful automated design tools. IC and software engineers collaborated during the definition of the third-generation LCA architecture to meet an important performance goal — an FPGA architecture and companion design tools for completely automatic placement and routing of 95% of all designs, plus a convenient way to complete the remaining few designs.

Configurable Logic Blocks

A number of architectural improvements contribute to the increased logic density and performance levels of the XC4000 families. The most important one is a more powerful and flexible CLB surrounded by a versatile set of routing resources, resulting in more "effective gates per CLB." The principal CLB elements are shown in Figure 1. Each new CLB also packs a pair of flip-flops and two independent 4-input function generators. The two function generators offer designers plenty of flexibility because most combinatorial logic functions need less than four inputs. Consequently, the design-software tools can deal with each function generator independently, thus improving cell usage.

Thirteen CLB inputs and four CLB outputs provide access to the function generators and flip-flops. More than double the number available in the XC3000 families, these inputs and outputs connect to the programmable interconnect resources outside the block. Four independent inputs are provided to each of two function generators (F1 – F4 and G1 – G4). These function generators, whose outputs are labeled F' and G', are each capable of implementing any arbitrarily defined Boolean function of their four inputs. The function generators are implemented as memory look-up tables; therefore, the propagation delay is independent of the function being implemented. A third function generator, labeled H', can implement any Boolean function of its three inputs: F' and G' and a third input from outside the block (H1). Signals from the function generators can exit the CLB on two outputs; F' or H' can be connected to the X output, and G' or H' can be connected to the Y output. Thus, a CLB can be used to implement any two independent functions of up-to-four variables, or any single function of five variables, or any function of four variables together with some functions of five variables , or it can implement even some functions of up to nine variables. Implementing wide functions in a single block reduces both the number of blocks required and the delay in the signal path, achieving both increased density and speed.

The two storage elements in the CLB are edge-triggered D-type flip-flops with common clock (K) and clock enable (EC) inputs. A third common input (S/R) can be programmed as either an asynchronous set or reset signal

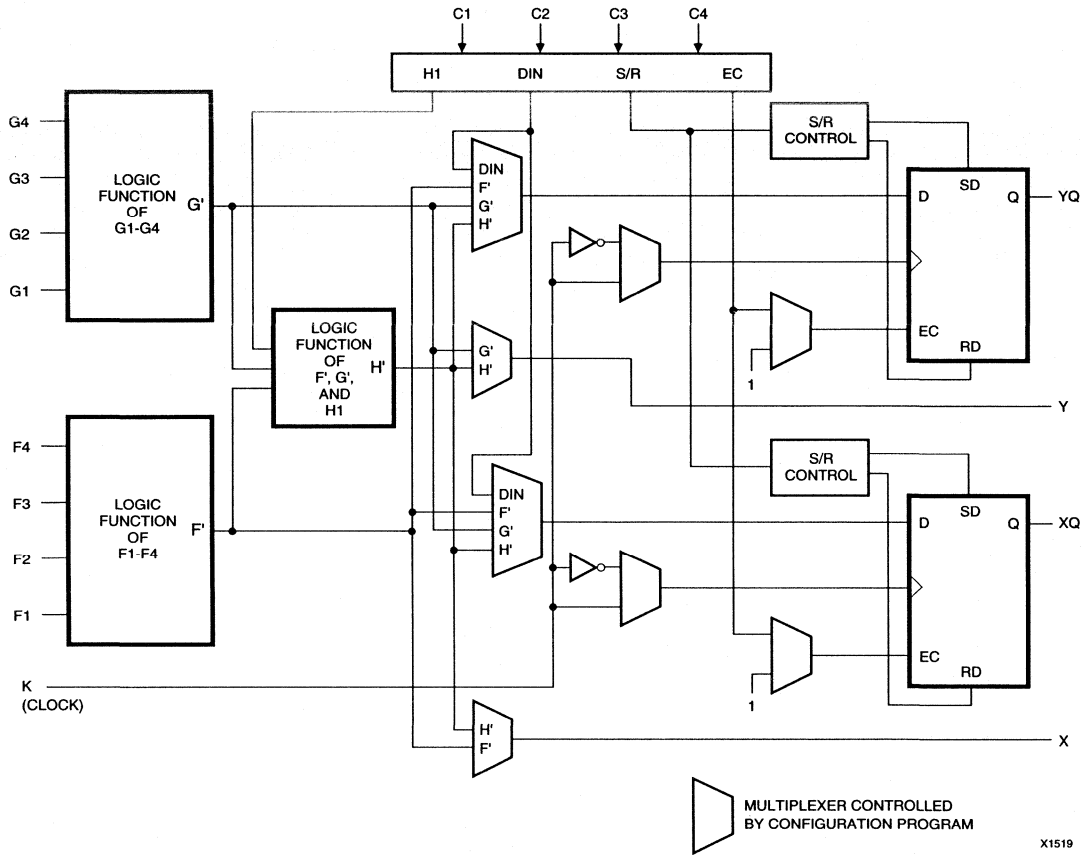


Figure 1. Simplified Block Diagram of XC4000-Families Configurable Logic Block

independently for each of the two registers; this input also can be disabled for either flip-flop. A separate global Set/Reset line (not shown in Figure 1) sets or clears each register during power-up, reconfiguration, or when a dedicated Reset net is driven active. This Reset net does not compete with other routing resources; it can be connected to any package pin as a global reset input.

Each flip-flop can be triggered on either the rising or falling clock edge. The source of a flip-flop data input is programmable: it is driven either by the functions F', G', and H', or the Direct In (DIN) block input . The flip-flops drive the XQ and YQ CLB outputs.

In addition, each CLB F' and G' function generator contains dedicated arithmetic logic for the fast generation of carry and borrow signals, greatly increasing the efficiency and performance of adders, subtracters, accumulators, comparators and even counters.

Multiplexers in the CLB map the four control inputs, labeled C1 through C4 in Figure 1, into the four internal control signals (H1, DIN, S/R, and EC) in any arbitrary manner.

The flexibility and symmetry of the CLB architecture facilitates the placement and routing of a given application. Since the function generators and flip-flops have independent inputs and outputs, each can be treated as a separate entity during placement to achieve high packing density. Inputs, outputs, and the functions themselves can freely swap positions within a CLB to avoid routing congestion during the placement and routing operation.

Speed Is Enhanced Two Ways

Delays in LCA-based designs are layout dependent. While this makes it hard to predict a worst-case guaranteed performance, there is a rule of thumb designers can consider — the system clock rate should not exceed one third to one half of the specified toggle rate. Critical portions of a design, shift registers and simple counters, can run faster — approximately two thirds of the specified toggle rate.

The XC4000 family can run at synchronous system clock rates of up to 60 MHz. This increase in performance over the previous families stems from two basic improvements: improved architecture and more abundant routing resources.

Improved Architecture

More Inputs: The versatility of the CLB function generators improves system speed significantly. Table 3 shows how the XC4000 families implement many functions more efficiently and faster than is possible with XC3000 devices. A 9-bit parity checker, for example, can be implemented in one CLB with a propagation delay of 7 ns. Using a XC3000-family device, the same function requires two CLBs with a propagation delay of 2 x 5.5 ns = 11 ns. One XC4000 CLB can determine whether two 4-bit words are identical, again with a 7-ns propagation delay. The ninth input can be used for simple ripple expansion of this identity comparator (25.5 ns over 16 bits, 51.5 ns over 32 bits), or a 2-layer identity comparator can generate the result of a 32-bit comparison in 15 ns, at the cost of a single extra CLB. Simpler functions like multiplexers also benefit from the greater flexibility of the XC4000-families CLB. A 16-input multiplexer uses 5 CLBs and has a delay of only 13.5 ns.

More Outputs: The CLB can pass the combinatorial output(s) to the interconnect network, but can also store the combinatorial result(s) or other incoming data in one or two flip-flops, and connect their outputs to the interconnect network as well. With XC3000-families CLBs the designer has to make a choice, either output the combinatorial function or the stored value. In the XC4000 families, the flip flops can be used as registers or shift registers without blocking the function generators from performing a different, perhaps unrelated task. This increases the functional density of the devices.

When a function generator drives a flip-flop in a CLB, the combinatorial propagation delay *overlaps completely* with the set-up time of the flip-flop. The set-up time is specified between the function generator inputs and the clock input. This represents a performance advantage over competing technologies where combinatorial delays must be added to the flip-flop set-up time.

Fast Carry: As described earlier, each CLB includes high-speed carry logic that can be activated by configuration. The two 4-input function generators can be configured as a 2-bit adder with built-in hidden carry that can be expanded to any length. This dedicated carry circuitry is so fast and efficient that conventional speed-up methods like carry generate/propagate are meaningless even at the 16-bit level, and of marginal benefit at the 32-bit level.

A 16-bit adder requires nine CLBs and has a combinatorial carry delay of 20.5 ns. Compare that to the 30 CLBs and 50 ns, or 41 CLBs and 30 ns in the XC3000 family.

The fast-carry logic opens the door to many new applications involving arithmetic operation, where the previous generations of FPGAs were not fast and/or not efficient enough. High-speed address offset calculations in microprocessor or graphics systems, and high-speed addition in digital signal processing are two typical applications.

Faster and More Efficient Counters: The XC4000-families fast-carry logic puts two counter bits into each CLB and runs them at a clock rate of up to 42 MHz for 16 bits, whether the counters are loadable or not. For a 16-bit

Table 3. Density and Performance for Several Common Circuit Functions

		XC3000 (-125)		XC4000 (-5)	
16-bit Decoder From Input Pad		15 ns	4 CLBs	12 ns	0 CLBs
24-bit Accumulator		17 MHz	46 CLBs	32 MHz	13 CLBs
State Machine Benchmark*		18 MHz	34 CLBs	30 MHz	26 CLBs
16:1 Multiplexer		16 ns	8 CLBs	16 ns	5 CLBs
16-bit Undirectional	Max Density	20 MHz	16 CLBs	40 MHz	8 CLBs
Loadable Counter	Max Speed	34 MHz	23 CLBs	42 MHz	9 CLBs
16-bit U/D Counter	Max Density	20 MHz	16 CLBs	40 MHz	8 CLBs
	Max Speed	30 MHz	27 CLBs	40 MHz	8 CLBs
16-bit Adder	Max Density	50 ns	30 CLBs	20.5 ns	9 CLBs
	Max Speed	30 ns	41 CLBs	20.5 ns	9 CLBs

* 16 states, 40 transitions, 10 inputs, 8 outputs

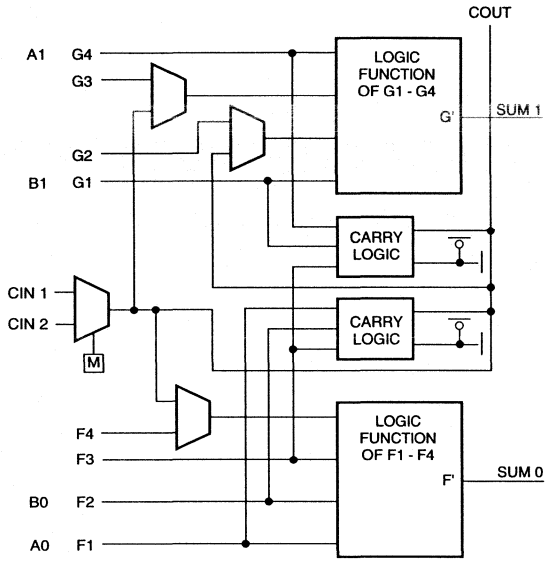


Figure 2. Fast Carry Logic in Each CLB

up/down counter, this means twice the speed in half the number of CLBs, compared with the XC3000 families.

Pipelining Speeds Up The System: The abundance of flip-flops in the CLBs invites pipelined designs. This is a powerful way of increasing performance by breaking the function into smaller subfunctions and executing them in parallel, passing on the results through pipeline flip-flops. This method should be seriously considered wherever total performance is more important than simple through-delay.

Wide Edge Decoding: For years, FPGAs have suffered from the lack of wide decoding circuitry. When the address or data field is wider than the function generator inputs (five bits in the XC3000 families), FPGAs need multi-level decoding and are thus slower than PALs. The XC4000-family CLBs have nine inputs; any decoder of up to nine inputs is, therefore, compact and fast. But, there is also a need for much wider decoders, especially for address decoding in large microprocessor systems. The XC4000 family has four programmable decoders located on each edge of each device. Each of these wired-AND gates is capable of accepting up to 42 inputs on the XC4005 and 72 on the XC4013. These decoders may also be split in two when a large number of narrower decoders are required for a maximum of 32 per device. These dedicated decoders accept I/O signals and internal signals as inputs and generate a decoded internal signal in 18 ns, pin-to-pin. The XC4000A family has only two decoder AND gates per edge which, when split provide a maximum of 16 per device. Very large PALs can be emulated by ORing the decoder outputs in a CLB. This decoding feature covers what has long been considered a weakness of FPGAs. Users often resorted to external PALs for simple but fast decoding functions. Now, the dedicated decoders in the XC4000 can implement these functions efficiently and fast.

Higher Output Current: The 4-mA maximum output current specification of today's FPGAs often forces the user to add external buffers, cumbersome especially on bidirectional I/O lines. The XC4000 families solve many of these problems by increasing the maximum output sink current to 12 mA. Two adjacent outputs may be interconnected to increase the output sink current to 24 mA. The FPGA can thus drive short buses on a pc board. The XC4000A and XC4000H outputs can sink 24 mA per output and can double up for 48 mA.

While the XC2000 and XC3000 families used complementary output transistors, the XC4000 outputs are n-channel for both pull-down and pull-up, somewhat analogous to the classical totem pole used in TTL. The reduced output High level (VOH) makes circuit delays more symmetrical for TTL-threshold systems. The XC4000H outputs have an optional p-channel output transistor.

Abundant Routing Resources

Connections between blocks are made by metal lines with programmable switching points and switching matrices. Compared to the previous LCA families, these routing resources have been increased dramatically.The number of globally distributed signals has been increased from two to eight, and these lines have access to any clock or logic input. The designer of synchronous systems can now distribute not only several clocks, but also control signals, all over the chip, without having to worry about any skew.

There are more than twice as many horizontal and vertical Longlines that can carry signals across the length or width of the chip with minimal delay and negligible skew.The horizontal Longlines can be driven by 3-state buffers, and can thus be used as unidirectional or bidirectional data buses; or they can implement wide multiplexers or wired-AND functions.

Single-length lines connect the switching matrices that are located at every intersection of a row and a column of CLBs. These lines provide the greatest interconnect flexibility, but cause a delay whenever they go through a switching matrix. Double-length lines bypass every other matrix, and provide faster signal routing over intermediate distances.

Compared to the XC3000 family, the XC4000 families have more than double the routing resources, and they are arranged in a far more regular fashion. In older devices,

inputs could not be driven by all adjacent routing lines. In the XC4000 families, these constraints have been largely eliminated. This makes it easier for the software to complete the routing of complex interconnect patterns.

Chip architects and software designers worked closely together to achieve a solution that is not only inherently powerful, but also easy to utilize by the software-driven design tools for Partitioning, Placement and Routing. The goal was to provide automated push-button software tools that complete almost all designs, even large and dense ones, automatically, without operator assistance. But these tools will still give the designer the option to get involved in the partitioning, placement and, to a lesser extent, even the routing of critical parts of the design, if that is needed to optimize the performance.

On-Chip Memory

The XC4000, XC4000A and XC4000H family devices are the first programmable logic devices with RAM accessible to the user.

An optional mode for each CLB makes the memory look-up tables in the F' and G' function generators usable as either a 16 x 2 or 32 x 1 bit array of Read/Write memory cells (Figure 3). The F1-F4 and G1-G4 inputs to the function generators act as address lines, selecting a particular memory cell in each look-up table. The functionality of the CLB control signals change in this configuration; the H1, DIN, and S/R lines become the two data inputs and the Write Enable (WE) input for the 16 x 2 memory. When the 32 x 1 configuration is selected, D1 acts as the fifth address bit and D0 is the data input. The contents of the memory cell(s) being addressed are available at the F' and G' function-generator outputs, and can exit the CLB through its X and Y outputs, or can be pipelined using the CLB flip-flop(s).

Configuring the CLB function generators as Read/Write memory does not affect the functionality of the other portions of the CLB, with the exception of the redefinition of the control signals. The H' function generator can be used to implement Boolean functions of F', G', and D1, and the D flip-flops can latch the F', G', H', or D0 signals.

The RAMs are very fast; read access is the same as logic delay, about 5.5 ns; write time is about 8 ns; both are several times faster than any off-chip solution. Such distributed RAM is a novel concept, creating new possibilities in system design: registered arrays of multiple accumulators, status registers, index registers, DMA counters, distributed shift registers, LIFO stacks, and FIFO buffers. The data path of a 16-byte FIFO uses four CLBs for storage, and six CLBs for address counting and multiplexing (Figure 4). With 32 storage locations per CLB, compared to two flip-flops per CLB, the cost of intelligent distributed memory has been reduced by a factor of 16.

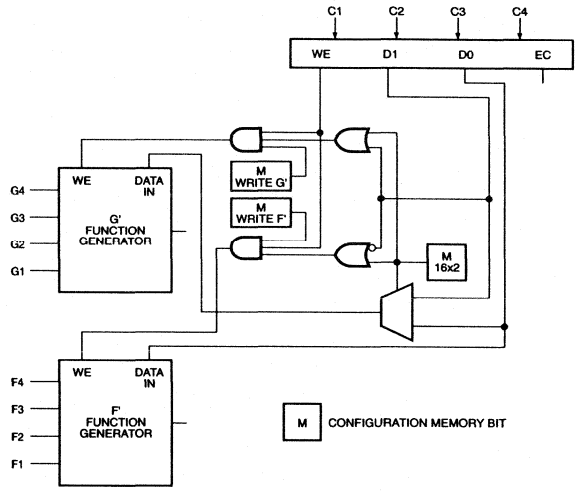


Figure 3. CLB Function Generators Can Be Used as Read/Write Memory Cells

Input/Output Blocks (IOBs), XC4000 and XC4000A Families

(for XC4000H family, see page 2-82)

User-configurable IOBs provide the interface between external package pins and the internal logic (Figure 5). Each IOB controls one package pin and can be defined for input, output, or bidirectional signals.

Two paths, labeled I1 and I2, bring input signals into the array. Inputs are routed to an input register that can be programmed as either an edge-triggered flip-flop or a level-sensitive transparent latch. Optionally, the data input to the register can be delayed by several nanoseconds to compensate for the delay on the clock signal, that first must

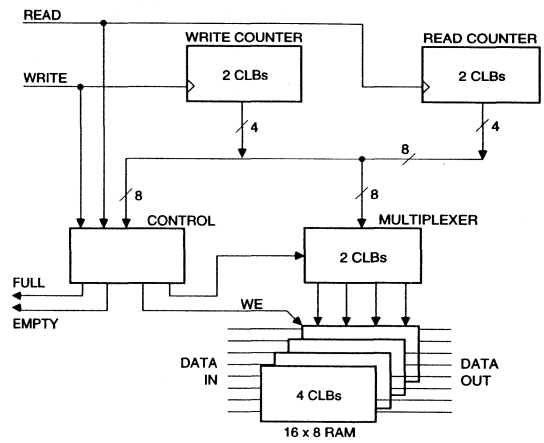


Figure 4. 16-byte FIFO

pass through a global buffer before arriving at the IOB. This eliminates the possibility of a data hold-time requirement at the external pin. The I1 and I2 signals that exit the block can each carry either the direct or registered input signal.

Output signals can be inverted or not inverted, and can pass directly to the pad or be stored in an edge-triggered flip-flop. Optionally, an output enable signal can be used to place the output buffer in a high-impedance state, implementing 3-state outputs or bidirectional I/O. Under configuration control, the output (OUT) and output enable (OE) signals can be inverted, and the slew rate of the output buffer can be reduced to minimize power bus transients when switching non-critical signals. Each XC4000-families output buffer is capable of sinking 12 mA; two adjacent output buffers can be wire-ANDed externally to sink up to 24 mA. In the XC4000A and XC4000H families, each output buffer can sink 24 mA.

There are a number of other programmable options in the IOB. Programmable pull-up and pull-down resistors are useful for tying unused pins to V_{CC} or ground to minimize power consumption. Separate clock signals are provided for the input and output registers; these clocks can be inverted, generating either falling-edge or rising-edge triggered flip-flops. As is the case with the CLB registers, a global set/reset signal can be used to set or clear the input and output registers whenever the RESET net is active.

Embedded logic attached to the IOBs contains test structures compatible with IEEE Standard 1149.1 for boundary-scan testing, permitting easy chip and board-level testing.

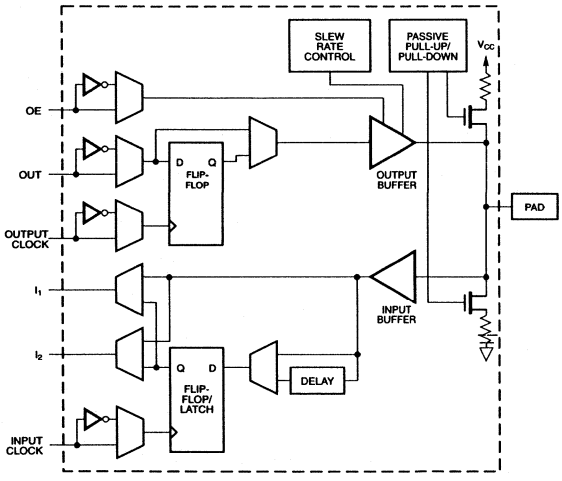


Figure 5. XC4000 and XC4000A Families Input/Output Block

Programmable Interconnect

All internal connections are composed of metal segments with programmable switching points to implement the desired routing. An abundance of different routing resources is provided to achieve efficient automated routing. The number of routing channels is scaled to the size of the array; i.e., it increases with array size.

In previous generations of LCAs, the logic-block inputs were located on the top, left, and bottom of the block; outputs exited the block on the right, favoring left-to-right data flow through the device. For the third-generation family, the CLB inputs and outputs are distributed on all four sides of the block, providing additional routing flexibility (Figure 6). In general, the entire architecture is more symmetrical and regular than that of earlier generations, and is more suited to well-established placement and routing algorithms developed for conventional mask- programmed gate-array design.

There are three main types of interconnect, distinguished by the relative length of their segments: single-length lines, double-length lines, and Longlines. Note: The number of routing channels shown in Figures 6 and 9 are for illustration purposes only; the actual number of routing channels varies with array size. The routing scheme was designed for minimum resistance and capacitance of the average routing path, resulting in significant performance improvements.

The single-length lines are a grid of horizontal and vertical lines that intersect at a Switch Matrix between each block. Figure 6 illustrates the single-length interconnect lines

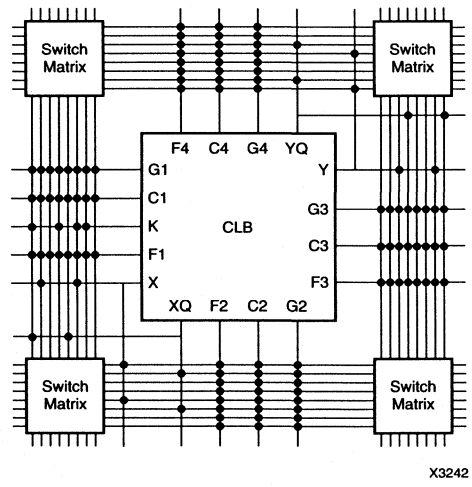


Figure 6. Typical CLB Connections to Adjacent Single-Length Lines

surrounding one CLB in the array. Each Switch Matrix consists of programmable n-channel pass transistors used to establish connections between the single-length lines (Figure 7). For example, a signal entering on the right side of the Switch Matrix can be routed to a single-length line on the top, left, or bottom sides, or any combination thereof, if multiple branches are required. Single-length lines are normally used to conduct signals within a localized area and to provide the branching for nets with fanout greater than one.

Compared to the previous generations of LCA architectures, the number of possible connections through the Switch Matrix has been reduced. This decreases capacitive loading and minimizes routing delays, thus increasing performance. However, a much more versatile set of connections between the single-length lines and the CLB inputs and outputs more than compensate for the reduction in Switch Matrix options, resulting in overall increased routability.

The function generator and control inputs to the CLB (F1-F4, G1-G4, and C1-C4) can be driven from any adjacent single-length line segment (Figure 6). The CLB clock (K) input can be driven from one-half of the adjacent single-length lines. Each CLB output can drive several of the single-length lines, with connections to both the horizontal and vertical Longlines.

The double-length lines (Figure 8) consist of a grid of metal segments twice as long as the single-length lines; i.e, a double-length line runs past two CLBs before entering a Switch Matrix. Double-length lines are grouped in pairs with the Switch Matrices staggered so that each line goes through a Switch Matrix at every other CLB location in that row or column. As with single-length lines, all the CLB inputs except K can be driven from any adjacent double-length line, and each CLB output can drive nearby double-length lines in both the vertical and horizontal planes. Double-length lines provide the most efficient implementation of intermediate length, point-to-point interconnections.

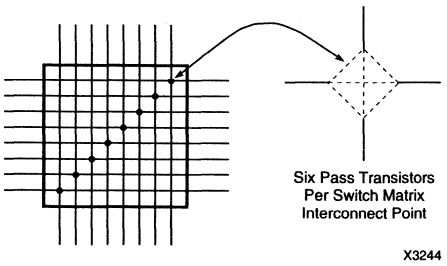


Figure 7. Switch Matrix

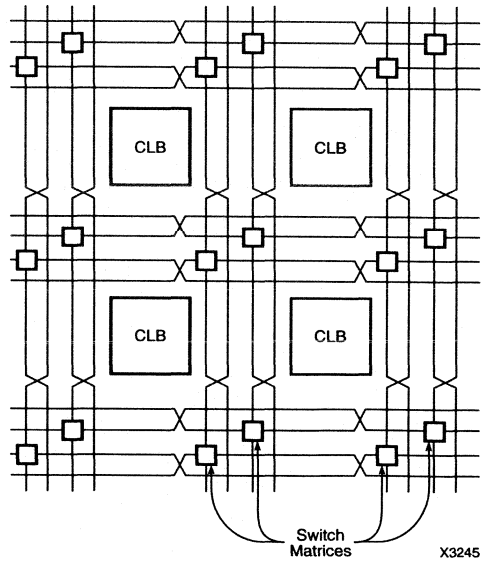


Figure 8. Double-Length Lines

Longlines form a grid of metal interconnect segments that run the entire length or width of the array (Figure 9). Additional vertical longlines can be driven by special global buffers, designed to distribute clocks and other high fanout control signals throughout the array with minimal skew. Longlines are intended for high fan-out, time-critical signal nets. Each Longline has a programmable splitter switch at its center, that can separate the line into two independent routing channels, each running half the width or height of the array. CLB inputs can be driven from a subset of the adjacent Longlines; CLB outputs are routed to the Longlines via 3-state buffers or the single-length interconnected lines.

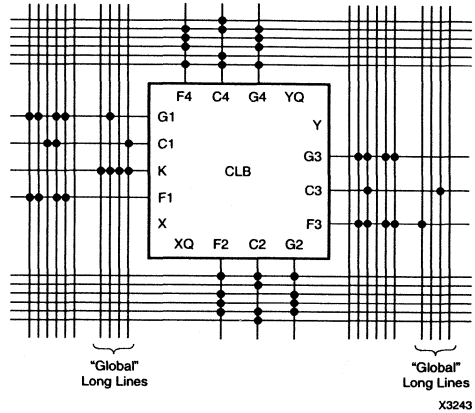


Figure 9. Longline Routing Resources with Typical CLB Connections

Communication between Longlines and single-length lines is controlled by programmable interconnect points at the line intersections. Double-length lines do not connect to other lines.

Three-State Buffers

A pair of 3-state buffers, associated with each CLB in the array, can be used to drive signals onto the nearest horizontal Longlines above and below the block. This feature is also available in the XC3000 generation of LCA devices. The 3-state buffer input can be driven from any X, Y, XQ, or YQ output of the neighboring CLB, or from nearby single-length lines; the buffer enable can come from nearby vertical single-length or Longlines. Another 3-state buffer with similar access is located near each I/O block along the right and left edges of the array. These buffers can be used to implement multiplexed or bidirectional buses on the horizontal Longlines. Programmable pull-up resistors attached to both ends of these Longlines help to implement a wide wired-AND function.

Special Longlines running along the perimeter of the array can be used to wire-AND signals coming from nearby IOBs or from internal Longlines.

Taking Advantage of Reconfiguration

LCA devices can be reconfigured to change logic function while resident in the system. This gives the system designer a new degree of freedom, not available with any other type of logic. Hardware can be changed as easily as software. Design updates or modifications are easy. An LCA device can even be reconfigured dynamically to perform different functions at different times. Reconfigurable logic can be used to implement system self diagnostics, create systems capable of being reconfigured for different environments or operations, or implement dual-purpose hardware for a given application. As an added benefit, use of reconfigurable LCA devices simplifies hardware design and debugging and shortens product time-to-market.

Development System

The powerful features of the XC4000 device families require an equally powerful, yet easy-to-use set of development tools. Xilinx provides an enhanced version of the Xilinx Automatic CAE Tools (XACT) optimized for the XC4000 families.

As with other logic technologies, the basic methodology for XC4000 FPGA design consists of three inter-related steps: entry, implementation, and verification. Popular 'generic' tools are used for entry and simulation (for example, Viewlogic System's ViewDraw schematic editor and ViewSim simulator), but architecture-specific tools are needed for implementation.

All Xilinx development system software is integrated under the Xilinx Design Manager (XDM), providing designers with a common user interface regardless of their choice of entry and verification tools. XDM simplifies the selection of command-line options with pull-down menus and on-line help text. Application programs ranging from schematic capture to Partitioning, Placement, and Routing (PPR) can be accessed from XDM, while the program-command sequence is generated and stored for documentation prior to execution. The XMAKE command, a design compilation utility, automates the entire implementation process, automatically retrieving the design's input files and performing all the steps needed to create configuration and report files.

Several advanced features of the XACT system facilitate XC4000 FPGA design. The MEMGEN utility, a memory compiler, implements on-chip RAM within an XC4000 FPGA. Relationally Placed Macros (RPMs) – schematic-based macros with relative locations constraints to guide their placement within the FPGA – help ensure an optimized implementation for common logic functions. XACT-Performance, a feature of the Partition, Place, and Route (PPR) implementation program, allows designers to enter their exact performance requirements during design entry, at the schematic level.

Design Entry

Designs can be entered graphically, using schematic-capture software, or in any of several text-based formats (such as Boolean equations, state-machine descriptions, and high-level design languages).

Xilinx and third-party CAE vendors have developed library and interface products compatible with a wide variety of design-entry and simulation environments. A standard interface-file specification, XNF (Xilinx Netlist File), is provided to simplify file transfers into and out of the XACT development system.

Xilinx offers XACT development system interfaces to the following design environments.

- Viewlogic Systems (ViewDraw, ViewSim)
- Mentor Graphics V7 and V8 (NETED, Quicksim, Design Architect, Quicksim II)
- OrCAD (SDT , VST)
- Synopsys (Design Compiler, FPGA Compiler)
- Xilinx-ABEL
- X-BLOX

Many other environments are supported by third-party vendors. Currently, more than 100 packages are supported.

The schematic library for the XC4000 FPGA reflects the wide variety of logic functions that can be implemented in these versatile devices. The library contains over 400 primitives and macros, ranging from 2-input AND gates to 16-bit accumulators, and including arithmetic functions,

comparators, counters, data registers, decoders, encoders, I/O functions, latches, Boolean functions, RAM and ROM memory blocks, multiplexers, shift registers, and barrel shifters.

Designing with macros is as easy as designing with standard SSI/MSI functions. The 'soft macro' library contains detailed descriptions of common logic functions, but does not contain any partitioning or routing information. The performance of these macros depends, therefore, on how the PPR software processes the design. Relationally Placed Macros (RPMs), on the other hand, do contain predetermined partitioning and relative placement information, resulting in an optimized implementation for these functions. Users can create their own library elements – either soft macros or RPMs – based on the macros and primitives of the standard library.

X-BLOX is a graphics-based high-level description language (HDL) that allows designers to use a schematic editor to enter designs as a set of generic modules. The X-BLOX compiler optimizes the modules for the target device architecture, automatically choosing the appropriate architectural resources for each function.

The XACT design environment supports hierarchical design entry, with top-level drawings defining the major functional blocks, and lower-level descriptions defining the logic in each block. The implementation tools automatically combine the hierarchical elements of a design. Different hierarchical elements can be specified with different design entry tools, allowing the use of the most convenient entry method for each portion of the design.

Design Implementation

The design implementation tools satisfy the requirement for an automated design process. Logic partitioning, block placement and signal routing, encompassing the design implementation process, are performed by the Partition, Place, and Route program (PPR). The partitioner takes the logic from the entered design and maps the logic into the architectural resources of the FPGA (such as the logic blocks, I/O blocks, 3-state buffers, and edge decoders). The placer then determines the best locations for the blocks, depending on their connectivity and the required performance. The router finally connects the placed blocks together. The PPR algorithms result in the fully automatic implementation of most designs. However, for demanding applications, the user may exercise various degrees of control over the automated implementation process. Optionally, user-designated partitioning, placement, and routing information can be specified as part of the design entry process. The implementation of highly-structured designs can greatly benefit from the basic floorplanning techniques familiar to designers of large gate arrays.

The PPR program includes XACT-Performance, a feature that allows designers to specify the timing requirements along entire paths during design entry. Timing path analysis routines in PPR then recognize and accommodate the user-specified requirements. Timing requirements can be entered on the schematic in a form directly relating to the system requirements (such as the targeted minimum clock frequency, or the maximum allowable delay on the data path between two registers). So, while the timing of each individual net is not predictable (nor does it need to be), the overall performance of the system along entire signal paths is automatically tailored to match user-generated specifications.

The automated implementation tools are complemented by the XACT Design Editor (XDE), an interactive graphics-based editor that displays a model of the actual logic and routing resources of the FPGA. XDE can be used to directly view the results achieved by the automated tools. Modifications can be made using XDE; XDE also performs checks for logic connectivity and possible design-rule violations.

Design Verification

The high development cost associated with common mask-programmed gate arrays necessitates extensive simulation to verify a design. Due to the custom nature of masked gate arrays, mistakes or last-minute design changes cannot be tolerated. A gate-array designer must simulate and test all logic and timing using simulation software. Simulation describes what happens in a system under worst-case situations. However, simulation is tedious and slow, and simulation vectors must be generated. A few seconds of system time can take weeks to simulate.

Programmable-gate-array users, however, can use in-circuit debugging techniques in addition to simulation. Because Xilinx devices are reprogrammable, designs can be verified in the system in real time without the need for extensive simulation vectors.

The XACT development system supports both simulation and in-circuit debugging techniques. For simulation, the system extracts the post-layout timing information from the design database. This data can then be sent to the simulator to verify timing-critical portions of the design. Back-annotation – the process of mapping the timing information back into the signal names and symbols of the schematic – eases the debugging effort.

For in-circuit debugging, XACT includes a serial download and readback cable (XChecker) that connects the device in the system to the PC or workstation through an RS232 serial port. The engineer can download a design or a design revision into the system for testing. The designer can also single-step the logic, read the contents of the numerous flip-flops on the device and observe internal logic levels. Simple modifications can be downloaded into the system in a matter of minutes.

The XACT system also includes XDelay, a static timing analyzer. XDelay examines a design's logic and timing to calculate the performance along signal paths, identify possible race conditions, and detect set-up and hold-time violations. Timing analyzers do not require that the user generate input stimulus patterns or test vectors.

Summary

The result of eight years of FPGA design experience and feedback from thousands of customers, the XC4000 families combine architectural versatility, on-chip RAM, increased speed and gate complexity with abundant routing resources and new, sophisticated software to achieve fully automated implementation of complex, high-performance designs.

7400 Equivalents

	# of CLBs
'138	5
'139	2
'147	5
'148	6
'150	5
'151	3
'152	3
'153	2
'154	16
'157	2
'158	2
'160	5
'161	6
'162	8
'163	8
'164	4
'165s	9
'166	5
'168	7
'174	3
'194	5
'195	3
'280	3
'283	8
'298	2
'352	2
'390	3
'518	3
'521	3

Barrel Shifters

brlshft4	4
brlshft8	13

4-Bit Counters

cd4ce	3
cd4cle	5
cd4rle	6
cb4ce	3
cb4cle	6
cb4re	5

8- and 16-Bit Counters

cb8ce	6
cb8re	10
cc16ce	10
cc16cle	11
cc16cled	21

Identity Comparators

comp4	1
comp8	2
comp16	5

Magnitude Comparators

compm4	4
compm8	9
compm16	20

Decoders

d2-4e	2
d3-8e	4
d4-16e	16

Multiplexers

m2-1e	1
m4-1e	1
m8-1e	3
m16-1e	5

Registers

rd4r	2
rd8r	4
rd16r	8

Shift Registers

sr8ce	4
sr16re	8

RAMs

ram 16x4	2

Explanation of counter nomenclature

cb = binary counter
cd = BCD counter
cc = cascadable binary counter
d = bidirectional
l = loadable
x = cascadable
e = clock enable
r = synchronous reset
c = asynchronous clear

Figure 10. CLB Count of Selected XC4000 Soft Macros

Detailed Functional Description

XC4000 and XC4000A Input/Output Blocks

(For XC4000H family, see page 2-82)

The IOB forms the interface between the internal logic and the I/O pads of the LCA device. Under configuration control, the output buffer receives either the logic signal (.out) routed from the internal logic to the IOB, or the complement of this signal, or this same data after it has been clocked into the output flip-flop.

As a configuration option, each flip-flop (CLB or IOB) is initialized as either set or reset, and is also forced into this programmable initialization state whenever the global Set/Reset net is activated after configuration has been completed. The clock polarity of each IOB flip-flop can be configured individually, as can the polarity of the 3-state control for the output buffer.

Each output buffer can be configured to be either fast or slew-rate limited, which reduces noise generation and ground bounce. Each I/O pin can be configured with either an internal pull-up or pull down resistor, or with no internal resistor. Independent of this choice, each IOB has a pull-up resistor during the configuration process.

The 3-state output driver uses a totem pole n-channel output structure. V_{OH} is one n-channel threshold lower than V_{CC}, which makes rise and fall delays more symmetrical.

Family	Per IOB Source	Per IOB Sink	Per IOB Pair Sink	# Slew Modes
XC4000	4	12	24	2
XC4000A	4	24	48	4
XC4000H	4	24*	48	2

*XC4000H devices can sink only 4 mA configured for SoftEdge mode

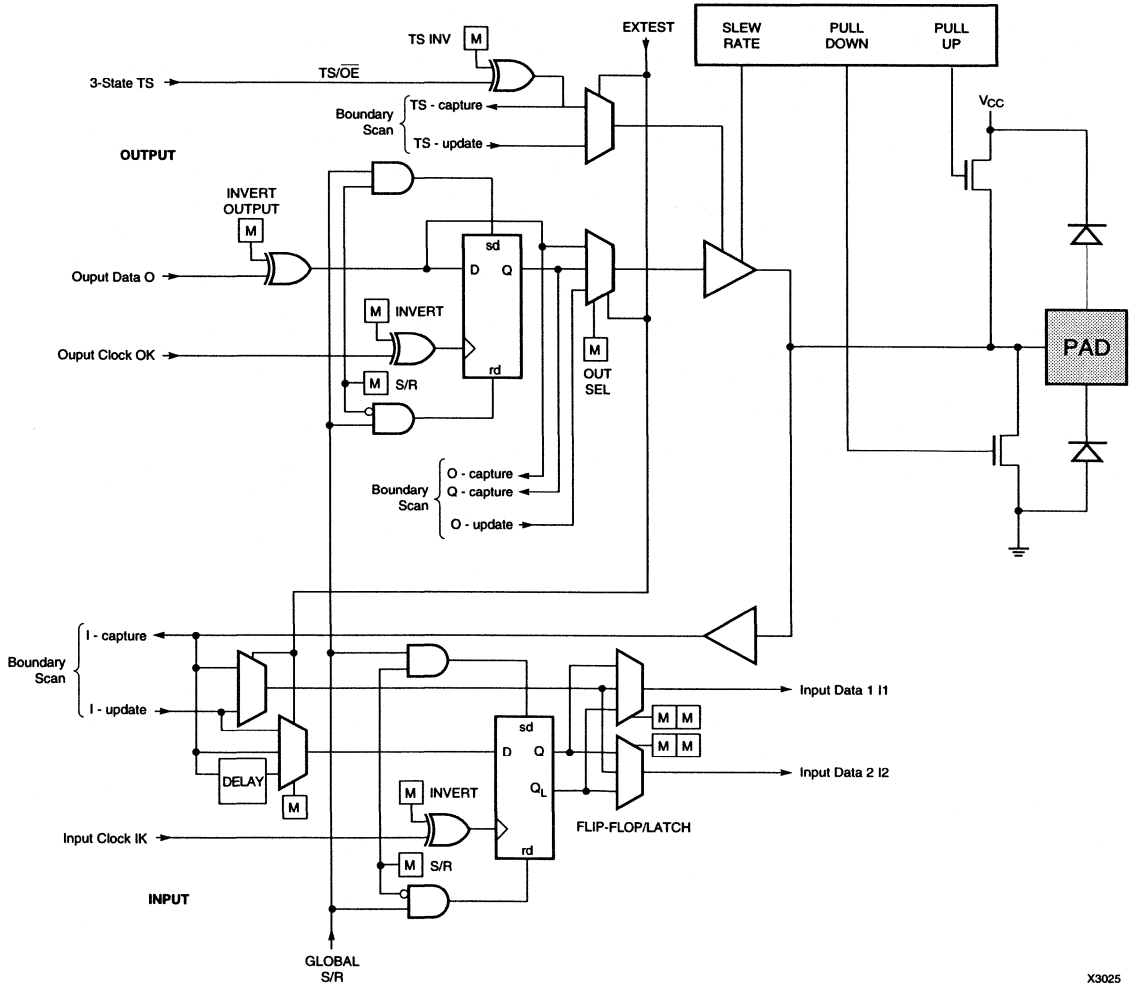


Figure 11. XC4000 and XC4000A I/O Block

The inputs drive TTL-compatible buffers with 1.2-V input threshold and a slight hysteresis of about 300 mV. These buffers drive the internal logic as well as the D-input of the input flip-flop.

Under configuration control, the set-up time of this flip-flop can be increased so that normal clock routing does not result in a hold-time problem. Note that the input flip-flop set-up time is defined between the data measured at the device I/O pin and the clock input at the IOB. Any clock routing delay must, therefore, be subtracted from this set-up time to arrive at the real set-up time requirement on the device pins. A short specified set-up time might, therefore, result in a negative set-up time at the device pins, i.e. a hold-time requirement, which is usually undesirable. The optional long set-up time can tolerate more clock delay without causing a hold-time requirement.

The input block has two connections to the internal logic, I1 and I2. Each of these is driven either by the incoming data, by the master or by the slave of the input flip-flop.

Wide Decoders

The periphery of the chip has four wide decoder circuits at each edge (two in the XC4000A). The inputs to each decoder are any of the I1 signals on that edge plus one local interconnect per CLB row or column. Each decoder generates High output (resistor pull-up) when the AND condition of the selected inputs, or their complements, is true. This is analogous to the AND term in typical PAL devices. Each decoder can be split at its center.

The decoder outputs can drive CLB inputs so they can be combined with other logic, or to form a PAL-like AND/OR structure. The decoder outputs can also be routed directly to the chip outputs. For fastest speed, the output should be on the same chip edge as the decoder.

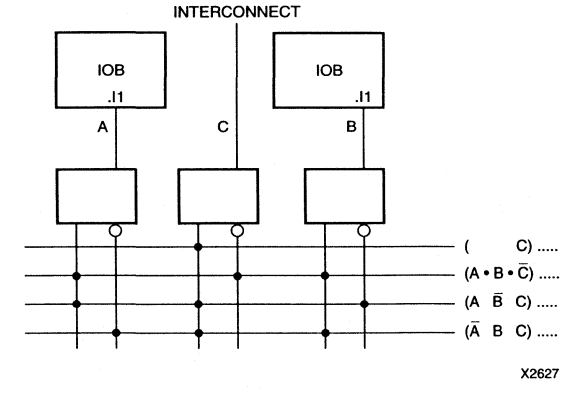


Figure 12. Example of Edge Decoding. Each row or column of CLBs provide up to three variables (or their complements)

Configurable Logic Blocks

Configurable Logic Blocks implement most of the logic in an LCA device. Two 4-input function generators (F and G) offer unrestricted versatility. A third function generator (H) can combine the outputs of F and G with a ninth input variable, thus implementing certain functions of up to nine variables, like parity check or expandable-identity comparison of two sets of four inputs.

The four control inputs C1 through C4 can each generate any one of four logic signals, used in the CLB.

- Enable Clock, Asynchronous Preset/Reset, DIN, and H1, when the memory function is disabled, or
- Enable Clock, Write Enable, D0, and D1, when the memory function is enabled.

Since the function-generator outputs are brought out independently of the flip-flop outputs, and DIN and H1 can be used as direct inputs to the two flip-flops, the two combinatorial and the two sequential functions in the CLB can be used independently. This versatility increases logic density and simplifies routing.

The asynchronous flip-flop input can be configured as either set or reset. This configuration option also determines the state in which the flip-flops become operational after configuration, as well as the effect of an externally or internally applied Set/Reset during normal operation.

Fast Carry Logic

The CLBs can generate the arithmetic-carry output for incoming operands, and can pass this extra output on to the next CLB function generator above or below. This connection is independent of normal routing resources and it is, presently, only supported by Hard Macros. A later software release will accomodate Soft Macros and will permit graphic editing of the fast logic circuitry. This fast carry logic is one of the most significant improvements in the XC4000 families, speeding up arithmetic and counting into the 60-MHz range.

Using Function Generators as RAMs

Using XC4000 devices, the designer can write into the latches that hold the configuration content of the function generators. Each function generator can thus be used as a small Read/Write memory, or RAM. The function generators in any CLB can be configured in three ways.

- Two 16 x 1 RAMs with two data inputs and two data outputs – identical or, if preferred, different addressing for each RAM
- One 32 x 1 RAM with one data input and one data output
- One 16 x 1 RAM plus one 5-input function generator

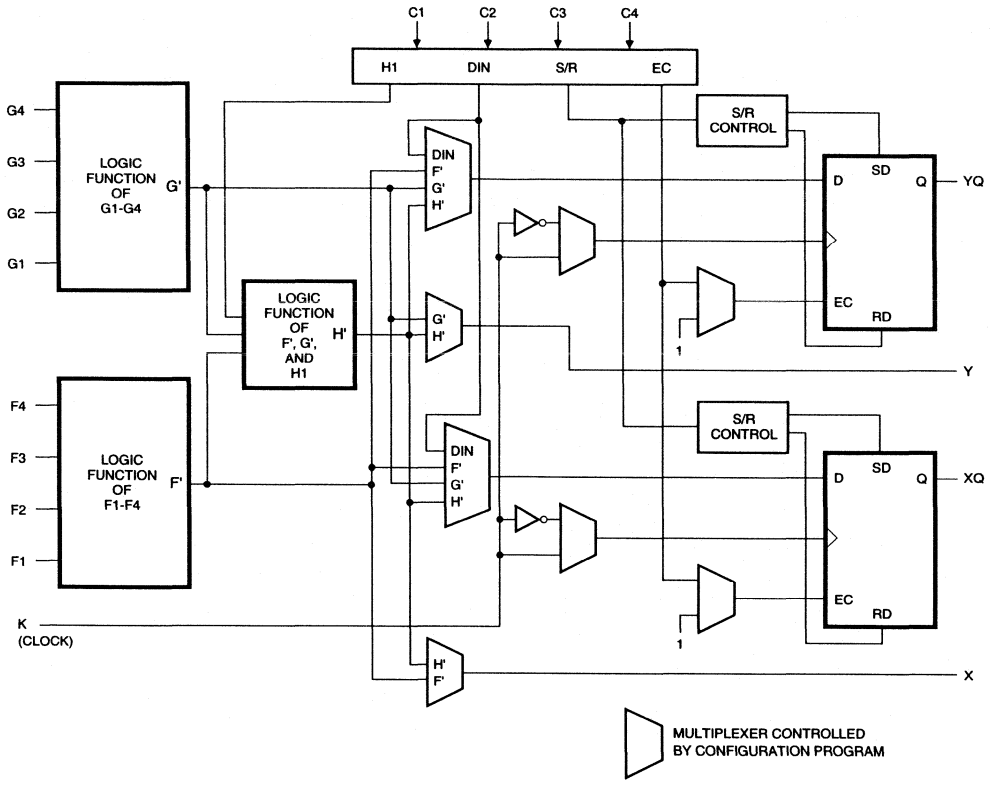


Figure 13. Simplified Block Diagram of XC4000 Configurable Logic Block

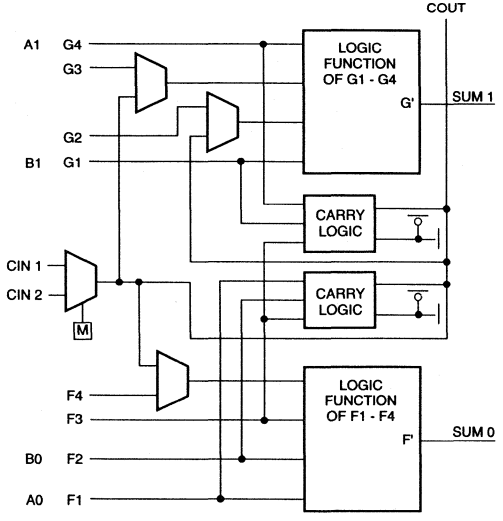


Figure 14. Fast Carry Logic in Each CLB

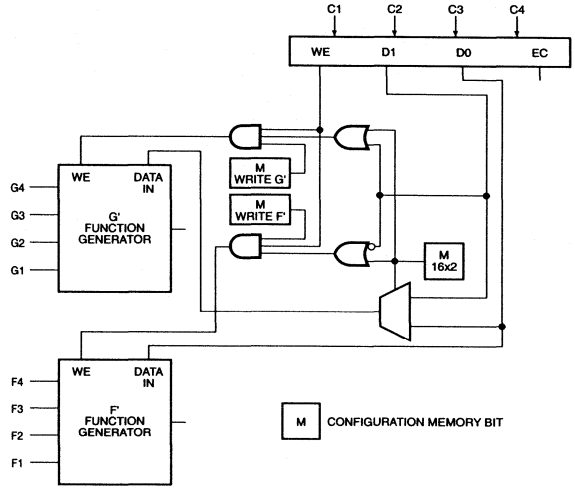


Figure 15. CLB Function Generators Can Be Used as Read/Write Memory Cells

Boundary Scan

Boundary Scan is becoming an attractive feature that helps sophisticated systems manufacturers test their PC boards more safely and more efficiently. The XC4000 family implements IEEE 1149.1-compatible BYPASS, PRELOAD/SAMPLE and EXTEST Boundary-Scan instructions. When the Boundary-Scan configuration option is selected, three normal user I/O pins become dedicated inputs for these functions.

The "bed of nails" has been the traditional method of testing electronic assemblies. This approach has become less appropriate, due to closer pin spacing and more sophisticated assembly methods like surface-mount technology and multi-layer boards. The IEEE Boundary Scan standard 1149.1 was developed to facilitate board-level testing of electronic assemblies. Design and test engineers can imbed a standard test logic structure in their electronic design. This structure is easily implemented with the serial and/or parallel connections of a four-pin interface on any Boundary-Scan-compatible IC. By exercising these signals, the user can serially load commands and data into these devices to control the driving of their outputs and to examine their inputs. This is an improvement over bed-of-nails testing. It avoids the need to overdrive device outputs, and it reduces the user interface to four pins. An optional fifth pin, a reset for the control logic, is described in the standard but is not implemented in the Xilinx part.

The dedicated on-chip logic implementing the IEEE 1149.1 functions includes a 16-state machine, an instruction register and a number of data registers. A register operation begins with a *capture* where a set of data is parallel loaded into the designated register for shifting out. The next state is *shift*, where captured data are shifted out while the desired data are shifted in. A number of states are provided for Wait operations. The last state of a register sequence is the *update* where the shifted content of the register is loaded into the appropriate instruction- or data-holding register, either for instruction-register decode or for data-register pin control.

The primary data register is the Boundary-Scan register. For each IOB pin in the LCA device, it includes three bits of shift register and three *update* latches for: in, out and 3-state control. Non-IOB pins have appropriate partial bit population for in or out only. Each Extest Capture captures all available input pins.

The other standard data register is the single flip-flop *bypass* register. It resynchronizes data being passed through a device that need not be involved in the current scan operation. The LCA device provides two user nets (BSCAN.SEL1 and BSCAN.SEL2) which are the decodes of two user instructions. For these instructions, two corresponding nets (BSCAN.TDO1 and BSCAN.TDO2) allow user scan data to be shifted out on TDO. The data register clock (BSCAN.DRCK) is available for control of test logic which the user may wish to implement with CLBs. The NAND of TCK and Run-test-idle is also provided (BSCAN.IDLE).

The XC4000 Boundary Scan instruction set also includes instructions to configure the device and read back the configuration data.

Table 4. Boundary Scan Instruction

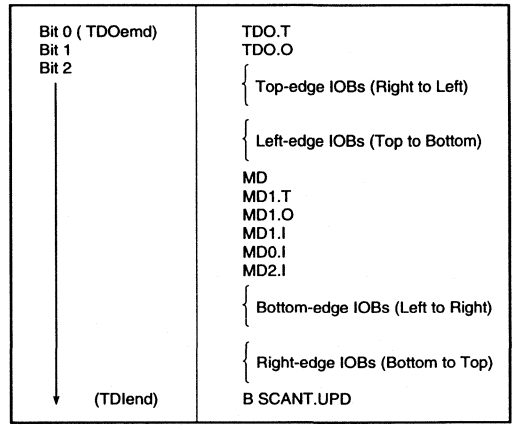
Instruction I_2	I_1	I_0	Test Selected	TDO Source	I/O Data Source
0	0	0	Extest	DR	DR
0	0	1	Sample/Preload	DR	Pin/Logic
0	1	0	User 1	TDO1	Pin/Logic
0	1	1	User 2	TDO2	Pin/Logic
1	0	0	Readback	Readback Data	Pin/Logic
1	0	1	Configure	DOUT	Disabled
1	1	0	Reserved	—	—
1	1	1	Bypass	Bypass Reg	Pin/Logic

X2679

Bit Sequence

The bit sequence within each IOB is: in, out, 3-state. From a cavity-up (XDE) view of the chip, starting in the upper right chip corner, the Boundary-Scan data-register bits have the following order.

Table 5. Boundary Scan Order



X2674

The data register also includes the following non-pin bits: TDO.T, and TDO.I, which are always bits 0 and 1 of the data register, respectively, and BSCANT.UPD which is always the last bit of the data register. These three Boundary-Scan bits are special-purpose Xilinx test signals. PROGRAM, CCLK and DONE are not included in the Boundary-Scan register. For more information regarding Boundary Scan, refer to XAPP 017.001, *Boundary Scan in XC4000 Devices*.

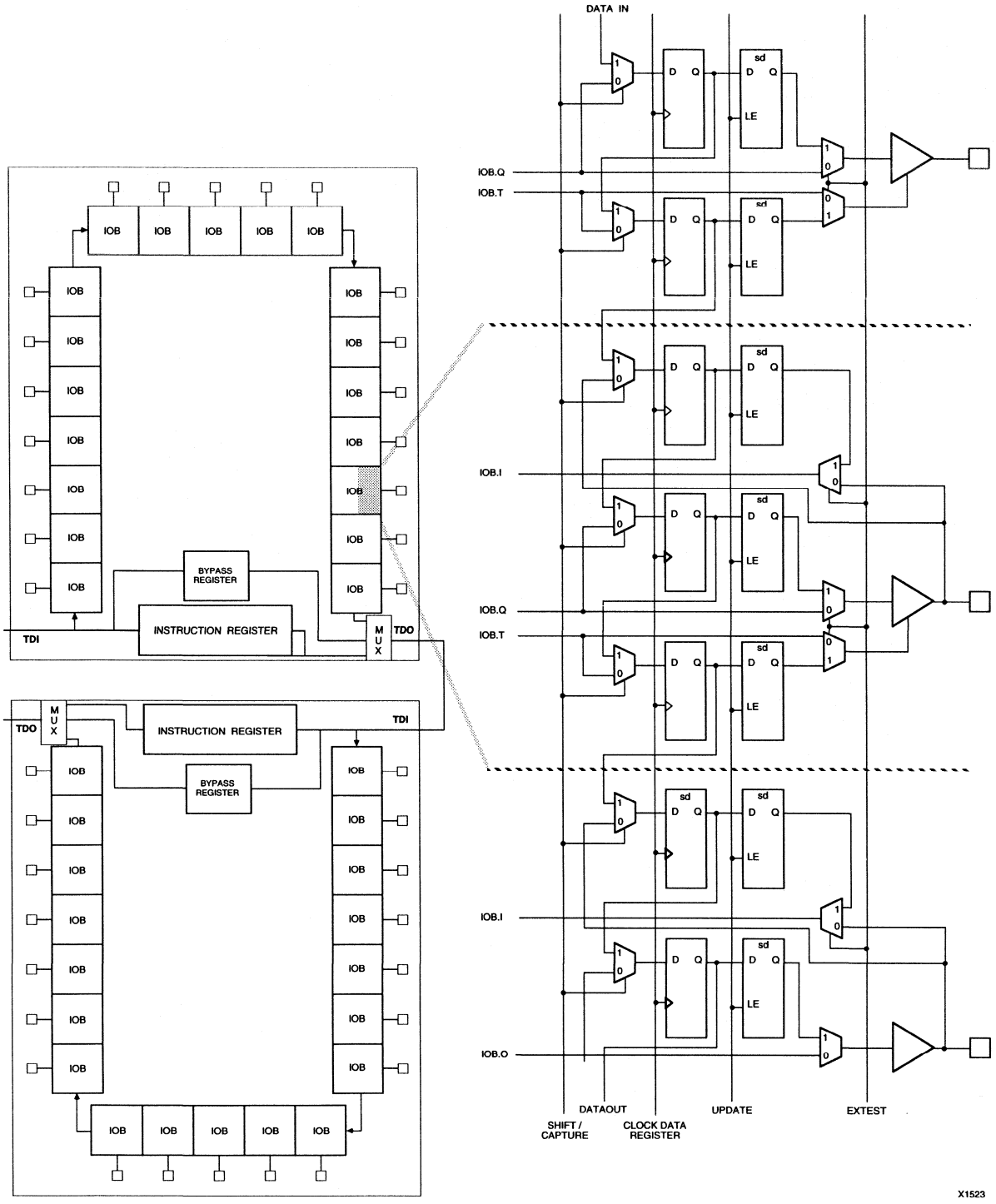


Figure 16. XC4000 Boundary Scan Logic. Includes three bits of Data Register per IOB, the IEEE 1149.1 Test Access Port controller, and the Instruction Register with decodes.

Interconnects

The XC4000 families use a hierarchy of interconnect resources.

- General purpose single-length and double-length lines offer fast routing between adjacent blocks, and highest flexibility for complex routes, but they incur a delay every time they pass through a switch matrix.
- Longlines run the width or height of the chip with negligible delay variations. They are used for signal distribution over long distances. Some Horizontal Longlines can be driven by 3-state or open-drain drivers, and can thus implement bidirectional buses or wired-AND decoding.
- Global Nets are optimized for the distribution of clock and time-critical or high-fan-out control signal. Four pad-driven Primary Global Nets offer shortest delay and negligible skew. Four pad-driven Secondary Global Nets have slightly longer delay and more skew due to heavier loading.

Each CLB column has four dedicated Vertical Longlines, each of these lines has access to a particular Primary Global Net, or to any one of the Secondary Global Nets. The Global Nets avoid clock skew and potential hold-time problems. The user must specify these Global Nets for all timing-sensitive global signal distribution.

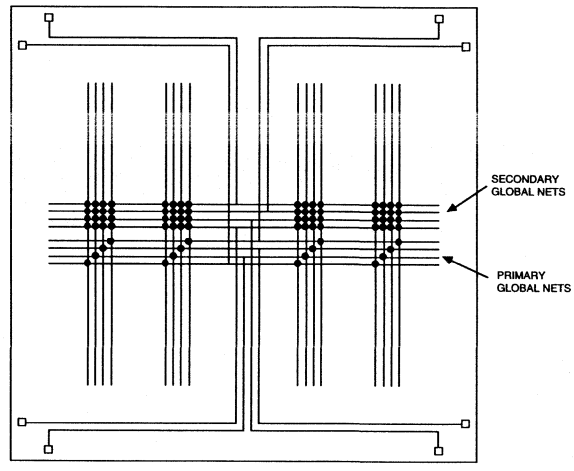
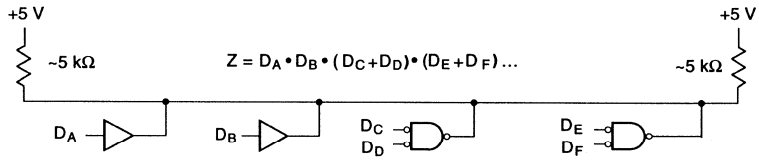
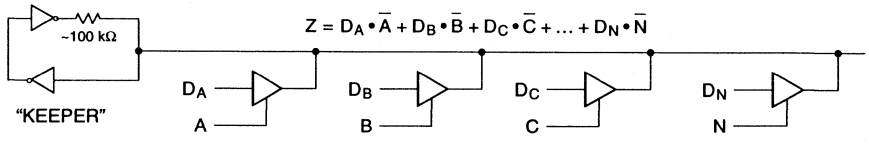


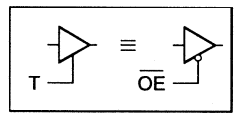
Figure 17. XC4000 Global Net Distribution. Four Lines per Column; Eight Inputs in the Four Chip Corners.



Open Drain Buffers Implement a Wired-AND Function. When all the buffer inputs are High the pull-up resistor(s) provide the High output.



3-State Buffers Implement a Multiplexer. The selection is accomplished by the buffer 3-state signal.



Active High T is Identical to Active Low Output Enable.

Figure 18. TBUFs Driving Horizontal Longlines.

Oscillator

An internal oscillator is used for clocking of the power-on time-out, configuration memory clearing, and as the source of CCLK in Master modes. This oscillator signal runs at a nominal 8 MHz and varies with process, V_{CC} and temperature between 10 MHz max and 4 MHz min. This signal is available on an output control net (OSCO) in the upper right corner of the chip, if the oscillator-run control bit is enabled in the configuration memory. Two of four resynchronized taps of the power-on time-out divider are also available on OSC1 and OSC2. These taps are at the fourth, ninth, fourteenth and nineteenth bits of the ripple divider. This can provide output signals of approximately 500 kHz,16 kHz, 490 Hz and 15 Hz.

Special Purpose Pins

The mode pins are sampled prior to configuration to determine the configuration mode and timing options. After configuration, these pins can be used as auxiliary connections: Mode 0 (MD0.I) and Mode 2 (MD2.I) as inputs and Mode 1 (MD1.O and MD1.T) as an output. The XACT development system will not use these resources unless they are explicitly specified in the design entry. These dedicated nets are located in the lower left chip corner and are near the readback nets. This allows convenient routing if compatibility with the XC2000 and XC3000 family conventions of M0/RT, M1/RD is desired.

Table 6. Configuration Modes

Mode	M2	M1	M0	CCLK	Data
Master Serial	0	0	0	output	Bit-Serial
Slave Serial	1	1	1	input	Bit-Serial
Master Parallel up	1	0	0	output	Byte-Wide, 00000 ↑
Master Parallel down	1	1	0	output	Byte-Wide, 3FFFF↓
Peripheral Synchr.	0	1	1	input	Byte-Wide
Peripheral Asynchr.	1	0	1	output	Byte-Wide
Reserved	0	1	0	—	—
Reserved	0	0	1	—	—
Peripheral Synchronous can be considered Slave Parallel					

Configuration

Configuration is the process of loading design-specific programming data into one or more LCA devices to define the functional operation of the internal blocks and their interconnections. This is somewhat like loading the command registers of a programmable peripheral chip. The XC4000 families use about 350 bits of configuration data per CLB and its associated interconnects. Each configuration bit defines the state of a static memory cell that controls either a function look-up table bit, a multiplexer input, or an interconnect pass transistor. The XACT development system translates the design into a netlist file. It automatically partitions, places and routes the logic and generates the configuration data in PROM format.

Modes

The XC4000 families have six configuration modes selected by a 3- bit input code applied to the M0, M1, and M2 inputs. There are three self-loading Master modes, two Peripheral modes and the Serial Slave mode used primarily for daisy-chained devices. During configuration, some of the I/O pins are used temporarily for the configuration process. See Table 6.

For a detailed description of these configuration modes, see pages 2-32 through 2-41.

Master

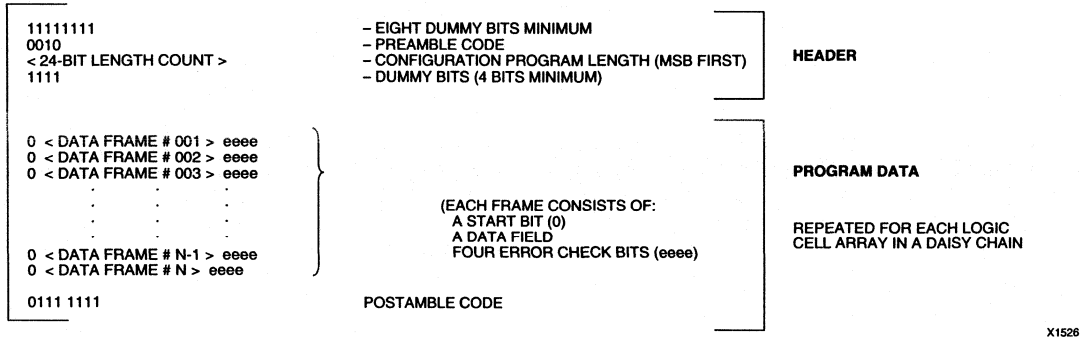
The Master modes use an internal oscillator to generate CCLK for driving potential slave devices, and to generate address and timing for external PROM(s) containing the configuration data. Master Parallel (up or down) modes generate the CCLK signal and PROM addresses and receive byte parallel data, which is internally serialized into the LCA data-frame format. The up and down selection generates starting addresses at either zero or 3FFFF, to be compatible with different microprocessor addressing conventions. The Master Serial mode generates CCLK and receives the configuration data in serial form from a Xilinx serial-configuration PROM.

Peripheral

The two Peripheral modes accept byte-wide data from a bus. A READY/BUSY status is available as a handshake signal. In the asynchronous mode, the internal oscillator generates a CCLK burst signal that serializes the byte-wide data. In the synchronous mode, an externally supplied clock input to CCLK serializes the data.

Serial Slave

In the Serial Slave mode, the LCA device receives serial-configuration data on the rising edge of CCLK and, after loading its configuration, passes additional data out, resynchronized on the next falling edge of CCLK. Multiple slave devices with identical configurations can be wired with parallel DIN inputs so that the devices can be configured simultaneously.



Device	XC4002A	XC4003A	XC4003/3H	XC4004A	XC4005A	XC4005/5H	XC4006	XC4008	XC4010/10D	XC4025
Gates	2,000	3,000	3,000	4,000	5000	5,000	6,000	8,000	10,000	25,000
CLBs (Row x Col)	64 (8 x 8)	100 (10 x 10)	100 (10 x 10)	144 (12 x 12)	196 (14 x 14)	196 (14 x 14)	256 (16 x 16)	324 (18 x 18)	400 (20 x 20)	1,024 (32 x 32)
IOBs	64	80	80/.160	96	112	112 (192)	128	144	160	256
Flip-flops	256	360	360/300	480	616	616 (392)	768	936	1,120	2,560
Horizontal TBUF Longlines	16	20	20	24	28	28	32	36	40	64
TBUFs/Longline	10	12	12	14	16	16	18	20	22	34
Bits per Frame	102	122	126	142	162	166	186	206	226	346
Frames	310	374	428	435	502	572	644	716	788	1,220
Program Data	31,628	45,636	53,936	62,204	81,332	94,960	119,792	147,504	178,096	422,128
PROM size (bits)	31,668	45,676	53,976	62,244	81,372	95,000	119,832	147,544	178,136	422,168

XC4000, 4000H: Bits per Frame = (10 x number of Rows) + 7 for the top + 13 for the bottom + 1 + 1 start bit + 4 error check bits
Number of Frames = (36 x number of Columns) + 26 for the left edge + 41 for the right edge + 1
XC4000A: Bits per Frame = (10 x number of Rows) + 6 for the top + 10 for the bottom + 1 + 1 start bit + 4 error check bits
Number of Frames = (32 x number of Columns) + 21 for the left edge + 32 for the right edge + 1
Program Data = (Bits per Frame x Number of Frames) + 8 postamble bits
PROM Size = Program Data + 40
The user can add more "one" bits as leading dummy bits in the header, or, if CRC = off, as trailing dummy bits at the end of any frame, following the four error check bits, but the Length Count value **must** be adjusted for all such extra "one" bits, even for leading extra ones at the beginning of the header.

Figure 19. Internal Configuration Data Structure.

Format

The configuration-data stream begins with a string of ones, a 0010 preamble code, a 24-bit length count, and a four-bit separator field of ones. This is followed by the actual configuration data in frames, each starting with a zero bit and ending with a four-bit error check. For each XC4XXX device, the MakeBits software allows a selection of CRC or non-CRC error checking. The non-CRC error checking tests for a 0110 end of frame field for each frame of a selected LCA device. For CRC error checking, MakeBits software calculates a running CRC of inserts a unique four-bit partial check at the end of each frame. The 11-bit CRC check of the last frame of an LCA device includes the last seven data bits. Detection of an error results in suspension of data loading and the pulling down of the $\overline{\text{INIT}}$ pin. In master modes, CCLK and address signals continue to operate externally. The user must detect $\overline{\text{INIT}}$ and initialize a new configuration by pulsing the $\overline{\text{PROGRAM}}$ pin or cycling V_{CC}. The length and number of frames depend on the device type. Multiple LCA devices can be connected in a daisy chain by wiring their CCLK pins in parallel and connecting the DOUT of each to the DIN of the next. The lead-master LCA device and following slaves each passes resynchronized configuration data coming from a single source. The Header data, including the length count, is passed through and is captured by each LCA

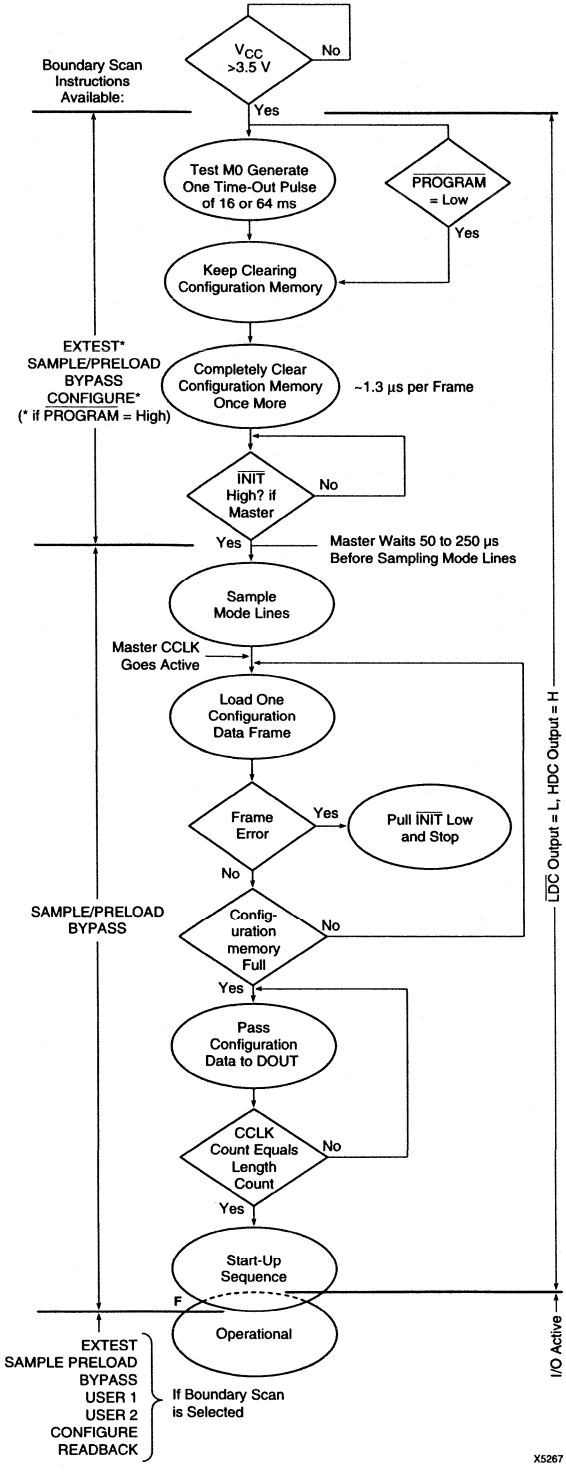


Figure 20. Start-up Sequence

device when it recognizes the 0010 preamble. Following the length-count data, any LCA device outputs a High on DOUT until it has received its required number of data frames.

After an LCA device has received its configuration data, it passes on any additional frame start bits and configuration data on DOUT. When the total number of configuration clocks applied after memory initialization equals the value of the 24-bit length count, the LCA device(s) begin the start-up sequence and become operational together.

Configuration Sequence

Configuration Memory Clear

When power is first applied or reapplied to an LCA device, an internal circuit forces initialization of the configuration logic. When V_{CC} reaches an operational level, and the circuit passes the write and read test of a sample pair of configuration bits, a nominal 16-ms time delay is started (four times longer when M0 is Low, i.e., in Master mode). During this time delay, or as long as the $\overline{\text{PROGRAM}}$ input is asserted, the configuration logic is held in a Configuration Memory Clear state. The configuration-memory frames are consecutively initialized, using the internal oscillator. At the end of each complete pass through the frame addressing, the power-on time-out delay circuitry and the level of the $\overline{\text{PROGRAM}}$ pin are tested. If neither is asserted, the logic initiates one additional clearing of the configuration frames and then tests the $\overline{\text{INIT}}$ input.

Initialization

During initialization and configuration, user pins HDC, $\overline{\text{LDC}}$ and $\overline{\text{INIT}}$ provide status outputs for system interface. The outputs, $\overline{\text{LDC}}$, $\overline{\text{INIT}}$ and DONE are held Low and HDC is held High starting at the initial application of power. The open drain $\overline{\text{INIT}}$ pin is released after the final initialization pass through the frame addresses. There is a deliberate delay of 50 to 250 µs before a Master-mode device recognizes an inactive $\overline{\text{INIT}}$. Two internal clocks after the $\overline{\text{INIT}}$ pin is recognized as High, the LCA device samples the three mode lines to determine the configuration mode. The appropriate interface lines become active and the configuration preamble and data can be loaded.

Configuration

The 0010 preamble code indicates that the following 24 bits represent the length count, i.e., the total number of configuration clocks needed to load the total configuration data. After the preamble and the length count have been passed through to all devices in the daisy chain, DOUT is held High to prevent frame start bits from reaching any daisy-chained devices. A specific configuration bit, early in the first frame of a master device, controls the configuration-clock rate and can increase it by a factor of eight. Each frame has a Low start bit followed by the frame-configura-

tion data bits and a 4-bit frame error field. If a frame data error is detected, the LCA device halts loading, and signals the error by pulling the open-drain $\overline{INIT}$ pin Low.

After all configuration frames have been loaded into an LCA device, DOUT again follows the input data so that the remaining data is passed on to the next device.

Start-Up

Start-up is the transition from the configuration process to the intended user operation. This means a change from one clock source to another, and a change from interfacing parallel or serial configuration data where most outputs are 3-stated, to normal operation with I/O pins active in the user-system. Start-up must make sure that the user-logic "wakes up" gracefully, that the outputs become active without causing contention with the configuration signals, and that the internal flip-flops are released from the global Reset or Set at the right time.

Figure 21 describes Start-up timing for the three Xilinx families in detail.

The **XC2000** family goes through a fixed sequence:

DONE goes High and the internal global Reset is de-activated one CCLK period after the I/O become active.

The **XC3000** family offers some flexibility: DONE can be programmed to go High one CCLK period before or after the I/O become active. Independent of DONE, the internal global Reset is de-activated one CCLK period before or after the I/O become active.

The **XC4000** family offers additional flexibility: The three events, DONE going High, the internal Reset/Set being de-activated, and the user I/O going active, can all occur in any arbitrary sequence, each of them one CCLK period before or after, or simultaneous with, any of the other.

The default option, and the most practical one, is for DONE to go High first, disconnecting the configuration data source and avoiding any contention when the I/Os become active one clock later. Reset/Set is then released another clock period later to make sure that user-operation starts from stable internal conditions. This is the most common sequence, shown with heavy lines in Figure 21, but the designer can modify it to meet particular requirements.

The XC4000 family offers another start-up clocking option: The three events described above don't have to be triggered by CCLK, they can, as a configuration option, be triggered by a user clock. This means that the device can wake up in synchronism with the user system.

The XC4000 family introduces an additional option: When this option is enabled, the user can externally hold the open-drain DONE output Low, and thus stall all further progress in the Start-up sequence, until DONE is released and has gone High. This option can be used to force synchronization of several LCA devices to a common user clock, or to guarantee that all devices are successfully configured before any I/Os go active.

Start-up Sequence

The Start-up sequence begins when the configuration memory is full, and the total number of configuration clocks received since INIT went High equals the loaded value of the length count. The next rising clock edge sets a flip-flop Q0 (see Figure 22), the leading bit of a 5-bit shift register.

The outputs of this register can be programmed to control three events.

- The release of the open-drain DONE output,
- The change of configuration-related pins to the user function, activating all IOBs.
- The termination of the global Set/Reset initialization of all CLB and IOB storage elements.

The DONE pin can also be wire-ANDed with DONE pins of other LCA devices or with other external signals, and can then be used as input to bit Q3 of the start-up register. This is called "Start-up Timing Synchronous to Done In" and labeled: CCLK_SYNC or UCLK_SYNC. When DONE is not used as an input, the operation is called Start-up Timing Not Synchronous to DONE In, and is labeled CCLK_NOSYNC or UCLK_NOSYNC. These labels are not intuitively obvious.

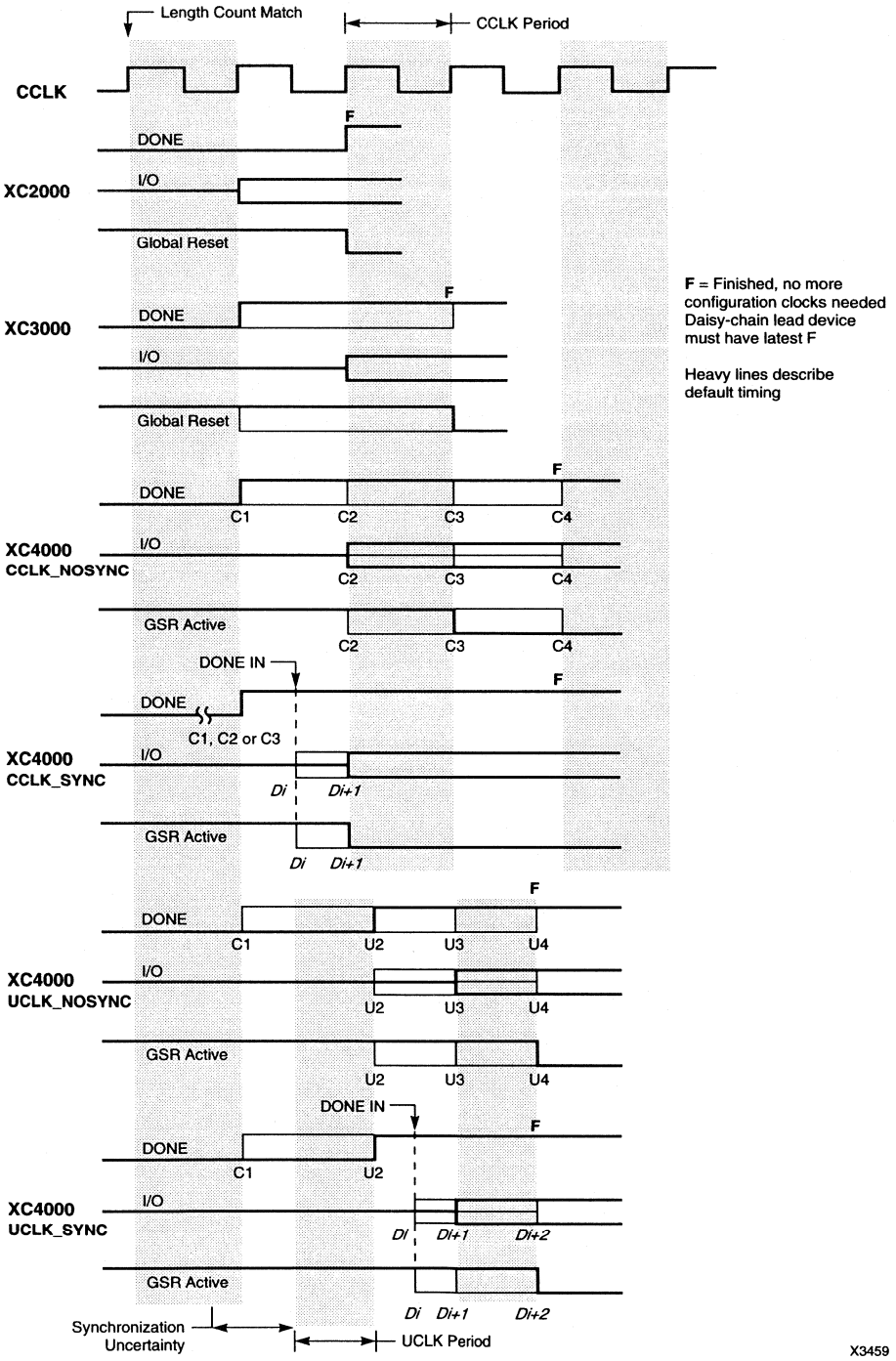
As a configuration option, the start-up control register beyond Q0 can be clocked either by subsequent CCLK pulses or from an on-chip user net called STARTUP.CLK.

Start-up from CCLK

If CCLK is used to drive the start-up, Q0 through Q3 provide the timing. Heavy lines in Figure 21 show the default timing which is compatible with XC2000 and XC3000 devices using early DONE and late Reset.The thin lines indicate all other possible timing options.

Start-up from a User Clock (STARTUP.CLK)

When, instead of CCLK, a user-supplied start-up clock is selected, Q1 is used to bridge the unknown phase relationship between CCLK and the user clock. This arbitration causes an unavoidable one-cycle uncertainty in the timing of the rest of the start-up sequence.



Note: Thick lines are default option.

Figure 21. Start-up Timing

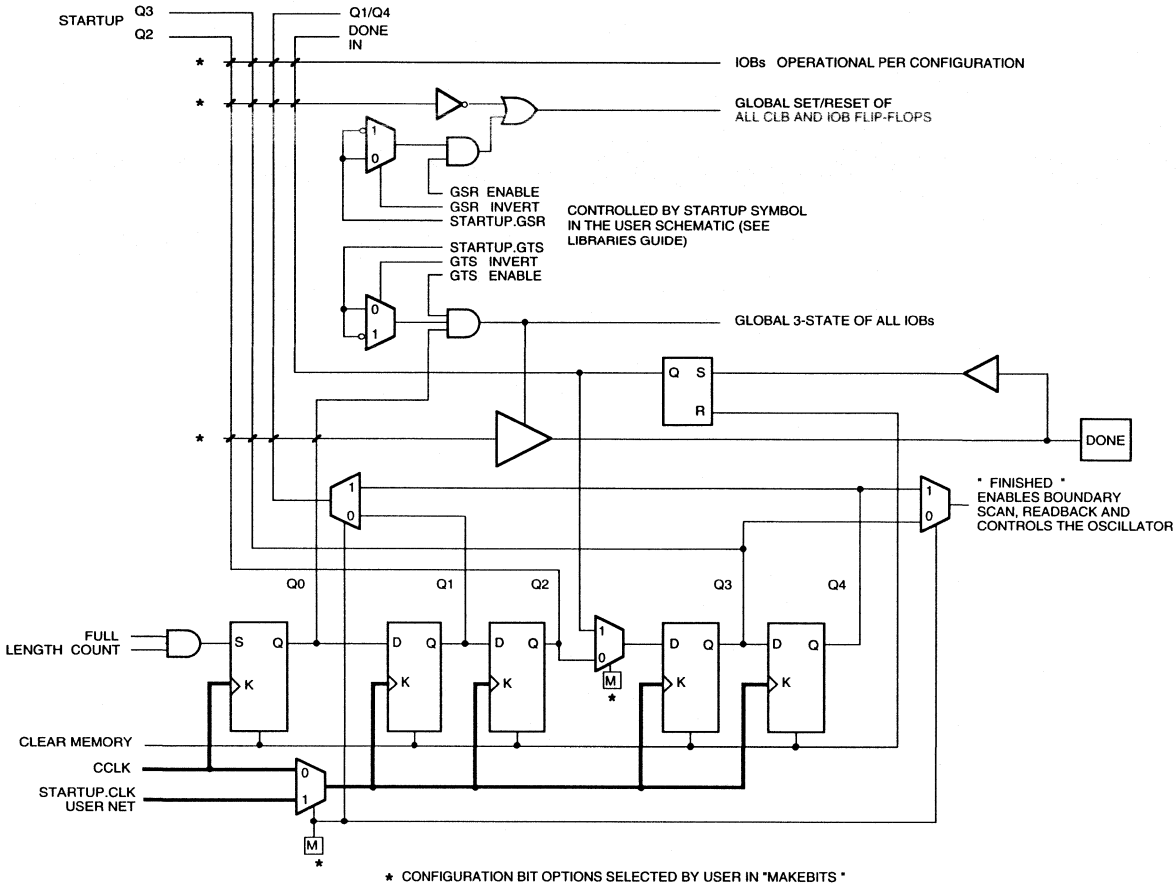


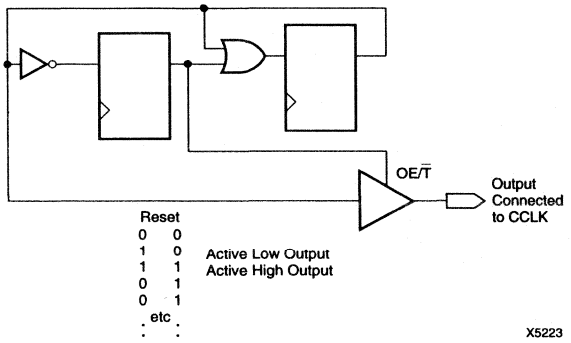
Figure 22. Start-up Logic

All Xilinx FPGAs of the XC2000, XC3000, XC4000 familiies use a compatible bitstream format and can, therefore, be connected in a daisy-chain in an arbitrary sequence. There is however one limitation. The lead device must belong to the highest family in the chain. If the chain contains XC4000 devices, the master cannot be an XC2000 or XC3000 device; if the daisy-chain contains XC3000 devices, the master cannot be an XC2000 device. The reason for this rule is shown in Figure 21 on the previous page. Since all devices in the chain store the same length count value and generate or receive one common sequence of CCLK pulses, they all recognize length-count match on the same CCLK edge, as indicated on the left edge of Figure 21. The master device will then drive additional CCLK pulses until it reaches its finish point F. The different families generate or require different numbers of additional CCLK pulses until they reach F.

Not reaching F means that the device does not really finish its configuration, although DONE may have gone High, the outputs became active, and the internal RESET was released. The user has some control over the relative timing of these events and can, therefore, make sure that they occur early enough.

But, for XC4000, not reaching F means that READBACK cannot be initiated and most Boundary Scan instructions cannot be used. This limitation has been critized by designers who want to use an inexpensive lead device in peripheral mode and have the more precious I/O pins of the XC4000 devices all available for user I/O. Here is a solution for that case.

One CLB and one IOB in the lead XC3000 device are used to generate the additional CCLK pulse required by the XC4000 devices. When the lead device removes the internal RESET signal, the 2-bit shift register responds to its clock input and generates an active Low output signal for the duration of the subsequent clock period. An external connection between this output and CCLK thus creates



the extra CCLK pulse. This solution requires one CLB, one IOB and pin, and an internal oscillator with a frequency of up to 5 MHz as available clock source. Obviously, this XC3000 master device must be configured with late Internal Reset, which happens to be the default option.

Using Global Set/Reset and Global 3-State Nets

The global Set/Reset (STARTUP.GSR) net can be driven by the user at any time to re-initialize all CLBs and IOBs to the same state they had at the end of configuration. For CLBs that is the same state as the one driven by the individually programmable asynchronous Set/Reset inputs. The global 3-state net (STARTUP.GTS), whenever activated after configuration is completed, forces all LCA outputs to the high-impedance state, unless Boundary Scan is enabled and is executing an EXTEST instruction.

Readback

The user can read back the content of configuration memory and the level of certain internal nodes without interfering with the normal operation of the device.

Readback reports not only the downloaded configuration bits, but can also include the present state of the device represented by the content of all used flip-flops and latches in CLBs and IOBs, as well as the content of function generators used as RAMs.

XC4000 Readback does not use any dedicated pins, but uses four internal nets (RDBK.TRIG, RDBK.DATA, RDBK.RIP and RDBK.CLK) that can be routed to any IOB.

After Readback has been initiated by a Low-to-High transition on RDBK.TRIG, the RDBK.RIP (Read In Progress) output goes High on the next rising edge of RDBK.CLK. Subsequent rising edges of this clock shift out Readback data on the RDBK.DATA net. Readback data does not include the preamble, but starts with five dummy bits (all High) followed by the Start bit (Low) of the first frame. The first two data bits of the first frame are always High.

Note that, in the XC4000 families, data is not inverted with respect to configuration the way it is in XC2000 and XC3000 families.

Each frame ends with four error check bits. They are read back as High. The last seven bits of the last frame are also read back as High. An additional Start bit (Low) and an 11-bit Cyclic Redundancy Check (CRC) signature follow, before RIP returns Low.

Readback options are: Read Capture, Read Abort, and Clock Select.

Read Capture

When the Readback Capture option is selected, the readback data stream includes sampled values of CLB and IOB signals imbedded in the data stream. The rising edge of RDBK.TRIG located in the lower-left chip corner, captures, in latches, the inverted values of the four CLB outputs and the IOB output flip-flops and the input signals I1, I2 . When the capture option is not selected, the values of the capture bits reflect the configuration data originally written to those memory locations. If the RAM capability of the CLBs is used, RAM data are available in readback, since they directly overwrite the F and G function-table configuration of the CLB.

Read Abort

When the Readback Abort option is selected, a High-to-Low transition on RDBK.TRIG terminates the readback operation and prepares the logic to accept another trigger. After an aborted readback, additional clocks (up-to-one readback clock per configuration frame) may be required to re-initialize the control logic. The status of readback is indicated by the output control net (RDBK.RIP).

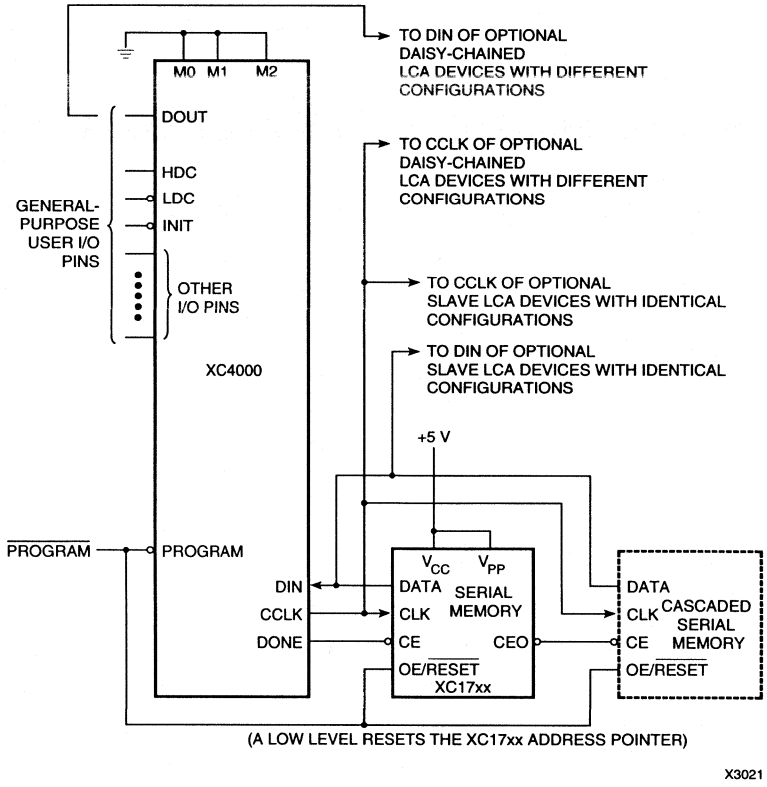
Clock Select

Readback control and data are clocked on rising edges of RDBK.CLK located in the lower right chip corner. CCLK is an optional clock. If Readback must be inhibited for security reasons, the readback control nets are simply not connected.

XChecker

The XChecker Universal Download/Readback Cable and Logic Probe uses the Readback feature for bitstream verification and for display of selected internal signals on the PC or workstation screen, effectively as a low-cost in-circuit emulator.

Master Serial Mode



In Master Serial mode, the CCLK output of the lead LCA device drives a Xilinx Serial PROM that feeds the LCA DIN input. Each rising edge of the CCLK output increments the Serial PROM internal address counter. This puts the next data bit on the SPROM data output, connected to the LCA DIN pin. The lead LCA device accepts this data on the subsequent rising CCLK edge.

The lead LCA device then presents the preamble data (and all data that overflows the lead device) on its DOUT pin. There is an internal pipeline delay of 1.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge. The user can specify Fast ConfigRate, which starting somewhere in the first frame, increases the CCLK frequency eight times, from a value between 0.5 and 1.25 MHz, to a value between 4 and 10 MHz. Note that most Serial PROMs are not compatible with this high frequency.

The SPROM $\overline{CE}$ input can be driven from either $\overline{LDC}$ or DONE. Using $\overline{LDC}$ avoids potential contention on the DIN pin, if this pin is configured as user-I/O, but $\overline{LDC}$ is then restricted to be a permanently High user output. Using DONE can also avoid contention on DIN, provided the early DONE option is invoked.

How to Delay Configuration After Power-Up

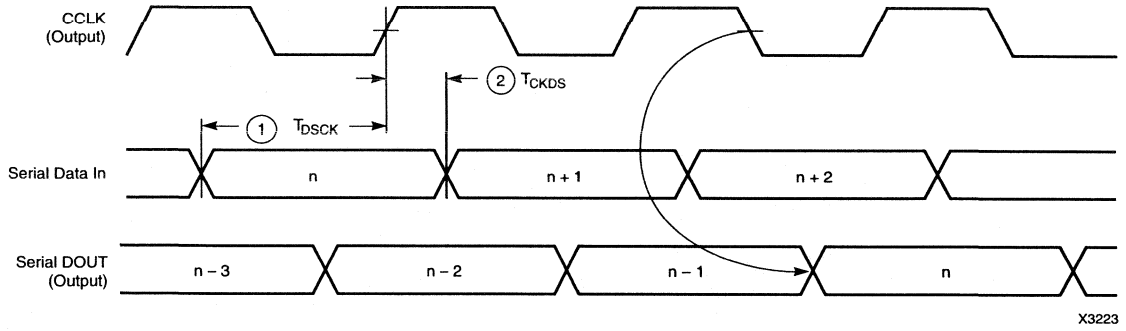
There are two methods to delay configuration after power-up: Put a logic Low on the $\overline{PROGRAM}$ input, or pull the bidirectional $\overline{INIT}$ pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27.)

A Low on the $\overline{PROGRAM}$ input is the more radical approach, and is recommended when the power-supply rise time is excessive or poorly defined. As long as $\overline{PROGRAM}$ is Low, the XC4000 device keeps clearing its configuration memory. When $\overline{PROGRAM}$ goes High, the configuration memory is cleared one more time, followed by the beginning of configuration, provided the INIT input is not externally held Low. Note that a Low on the $\overline{PROGRAM}$ input automatically forces a Low on the $\overline{INIT}$ output.

Using an open-collector or open-drain driver to hold $\overline{INIT}$ Low before the beginning of configuration, causes the LCA device to wait after having completed the configuration memory clear operation. When $\overline{INIT}$ is no longer held Low

externally, the device determines its configuration mode by capturing its status inputs, and is ready to start the configuration process. A master device waits an additional up to 250 µs to make sure that all slaves in the potential daisy-chain have seen $\overline{INIT}$ being High.

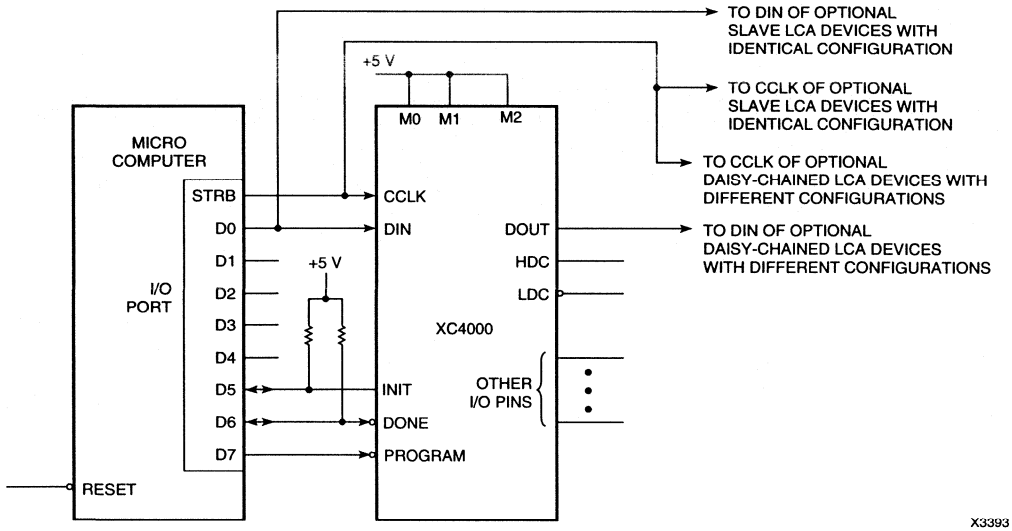
Master Serial Mode Programming Switching Characteristics



	Description	Symbol	Min	Max	Units
CCLK	Data In setup	1 T_{DSCK}	20		ns
	Data In hold	2 T_{CKDS}	0		ns

Notes: 1. At power-up, V_{CC} must rise from 2.0 V to Vcc min in less than 25 ms, otherwise delay configuration by pulling $\overline{PROGRAM}$ Low until V_{CC} is valid.
2. Configuration can be controlled by holding $\overline{INIT}$ Low with or until after the $\overline{INIT}$ of all daisy-chain slave mode devices is High.
3. Master-serial-mode timing is based on testing in slave mode.

Slave Serial Mode



In Slave Serial mode, an external signal drives the CCLK input(s) of the LCA device(s). The serial configuration bitstream must be available at the DIN input of the lead LCA device a short set-up time before each rising CCLK edge. The lead LCA device then presents the preamble data (and all data that overflows the lead device) on its DOUT pin.

There is an internal delay of 0.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

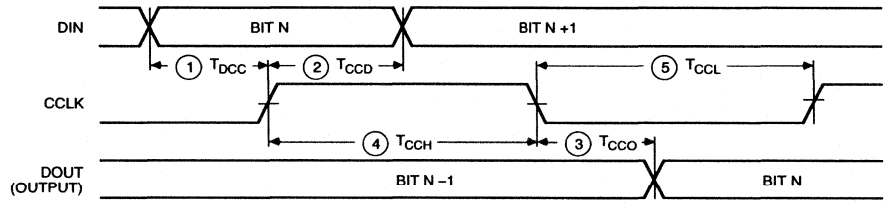
How to Delay Configuration After Power-Up

There are two methods to delay configuration after power-up: Put a logic Low on the $\overline{\text{PROGRAM}}$ input, or pull the bidirectional $\overline{\text{INIT}}$ pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27.)

A Low on the $\overline{\text{PROGRAM}}$ input is the more radical approach, and is recommended when the power-supply rise time is excessive or poorly defined. As long as $\overline{\text{PROGRAM}}$ is Low, the XC4000 device keeps clearing its configuration memory. When $\overline{\text{PROGRAM}}$ goes High, the configuration memory is cleared one more time, followed by the beginning of configuration, provided the INIT input is not externally held Low. Note that a Low on the $\overline{\text{PROGRAM}}$ input automatically forces a Low on the $\overline{\text{INIT}}$ output.

Using an open-collector or open-drain driver to hold $\overline{\text{INIT}}$ Low before the beginning of configuration, causes the LCA device to wait after having completed the configuration memory clear operation. When $\overline{\text{INIT}}$ is no longer held Low externally, the device determines its configuration mode by capturing its status inputs, and is ready to start the configuration process. A master device waits an additional max 250 µs to make sure that all slaves in the potential daisy-chain have seen $\overline{\text{INIT}}$ being High.

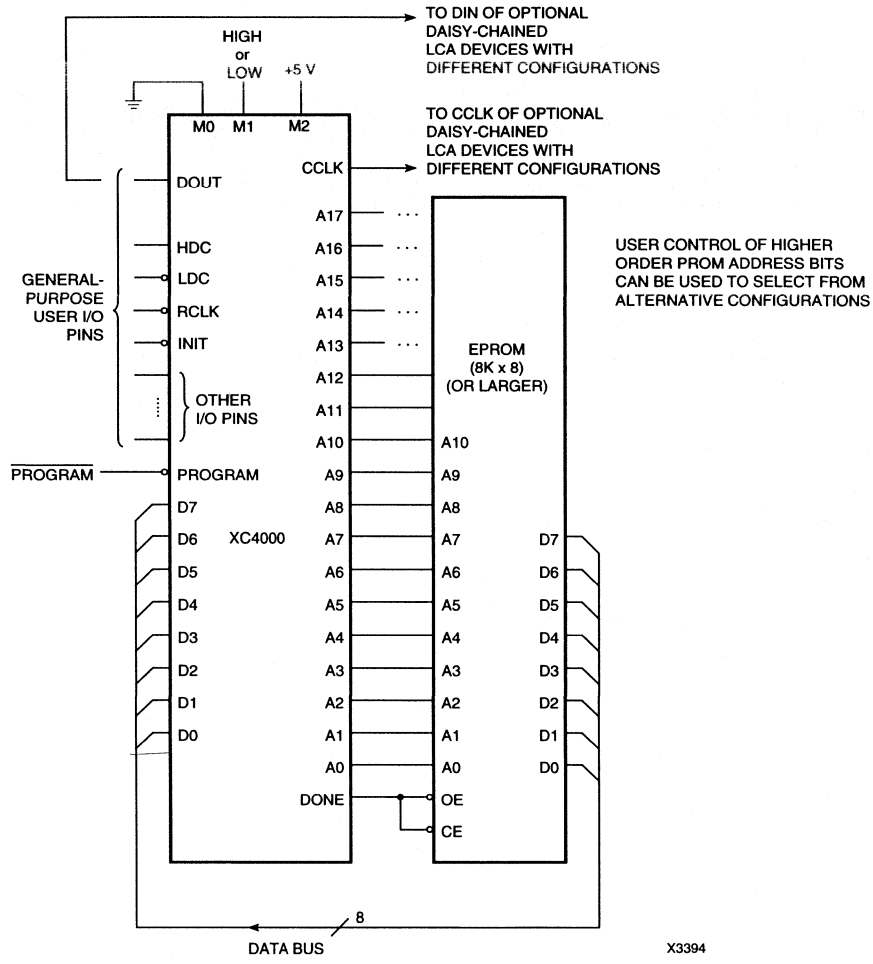
Slave Serial Mode Programming Switching Characteristics



	Description	Symbol		Min	Max	Units
CCLK	DIN setup	1	T_{DCC}	20		ns
	DIN hold	2	T_{CCD}	0		ns
	to DOUT	3	T_{CCO}		30	ns
	High time	4	T_{CCH}	45		ns
	Low time	5	T_{CCL}	45		ns
	Frequency		F_{CC}		10	MHz

Note: Configuration must be delayed until the $\overline{INIT}$ of all daisy-chained LCA devices is High.

Master Parallel Mode



In Master Parallel mode, the lead LCA device directly addresses an industry-standard byte-wide EPROM, and accepts eight data bits right before incrementing (or decrementing) the address outputs.

The eight data bits are serialized in the lead LCA device, which then presents the preamble data (and all data that overflows the lead device) on the DOUT pin. There is an internal delay of 1.5 CCLK periods, after the rising CCLK edge that accepts a byte of data (and also changes the EPROM address) until the falling CCLK edge that makes the LSB (D0) of this byte appear at DOUT. This means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

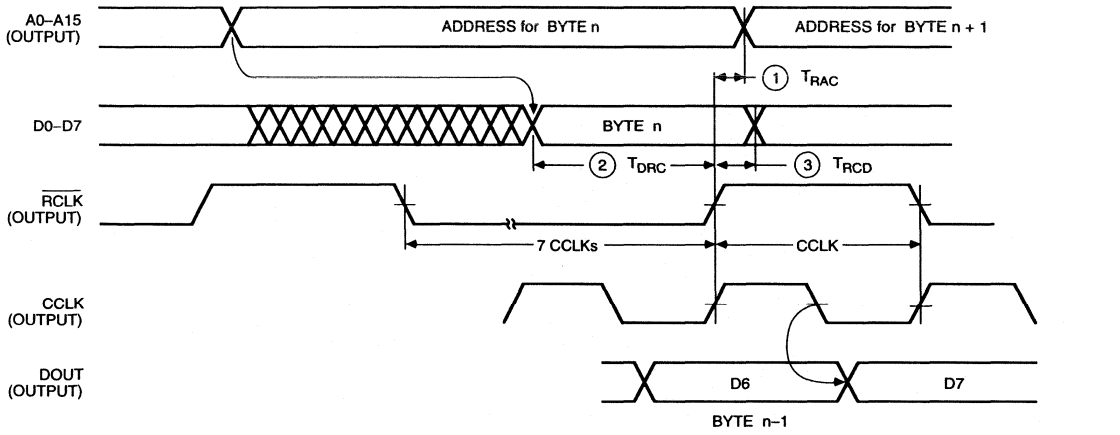
How to Delay Configuration After Power-Up

There are two methods to delay configuration after power-up: Put a logic Low on the $\overline{\text{PROGRAM}}$ input, or pull the bidirectional $\overline{\text{INIT}}$ pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27).

A Low on the $\overline{\text{PROGRAM}}$ input is the more radical approach, and is recommended when the power-supply rise time is excessive or poorly defined. As long as $\overline{\text{PROGRAM}}$ is Low, the XC4000 device keeps clearing its configuration memory. When $\overline{\text{PROGRAM}}$ goes High, the configuration memory is cleared one more time, followed by the beginning of configuration, provided the INIT input is not externally held Low. Note that a Low on the $\overline{\text{PROGRAM}}$ input automatically forces a Low on the $\overline{\text{INIT}}$ output.

Using an open-collector or open-drain driver to hold $\overline{INIT}$ Low before the beginning of configuration, causes the LCA device to wait after having completed the configuration memory clear operation. When $\overline{INIT}$ is no longer held Low externally, the device determines its configuration mode by capturing its status inputs, and is ready to start the configuration process. A master device waits an additional max 250 µs to make sure that all slaves in the potential daisy-chain have seen $\overline{INIT}$ being High.

Master Parallel Mode Programming Switching Characteristics



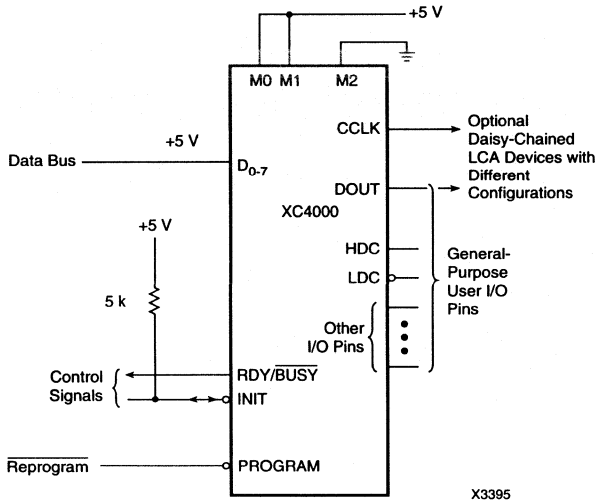
1105 30

	Description	Symbol		Min	Max	Units
RCLK	Delay to Address valid	1	T_{RAC}	0	200	ns
	Data setup time	2	T_{DRC}	60		ns
	Data setup time	3	T_{RCD}	0		ns

Notes: 1. At power-up, V_{CC} must rise from 2.0 V to Vcc min in less than 25 ms, otherwise delay configuration using $\overline{PROGRAM}$ until V_{CC} is valid.
2. Configuration can be delayed by holding $\overline{INIT}$ Low with or until after the $\overline{INIT}$ of all daisy-chain slave mode devices is High.
3. The first Data byte is loaded and CCLK starts at the end of the first $\overline{RCLK}$ active cycle (rising edge).

This timing diagram shows that the EPROM requirements are extremely relaxed: EPROM access time can be longer than 500 ns. EPROM data output has no hold-time requirements.

Synchronous Peripheral Mode



Synchronous Peripheral mode can also be considered Slave Parallel mode. An external signal drives the CCLK input(s) of the LCA device(s). The first byte of parallel configuration data must be available at the D inputs of the lead LCA device a short set-up time before each rising CCLK edge. Subsequent data bytes are clocked in on every eighth consecutive rising CCLK edge. The same CCLK edge that accepts data, also causes the RDY/BUSY output to go High for one CCLK period. The pin name is a misnomer. In Synchronous Peripheral mode it is really an ACKNOWLEDGE signal. Synchronous operation does not require this response, but it is a meaningful signal for test purposes.

The lead LCA device serializes the data and presents the preamble data (and all data that overflows the lead device) on its DOUT pin. There is an internal delay of 1.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge. In order to complete the serial shift operation, 10 additional CCLK rising edges are required after the last data byte has been loaded, plus one more CCLK cycle for each daisy-chained device.

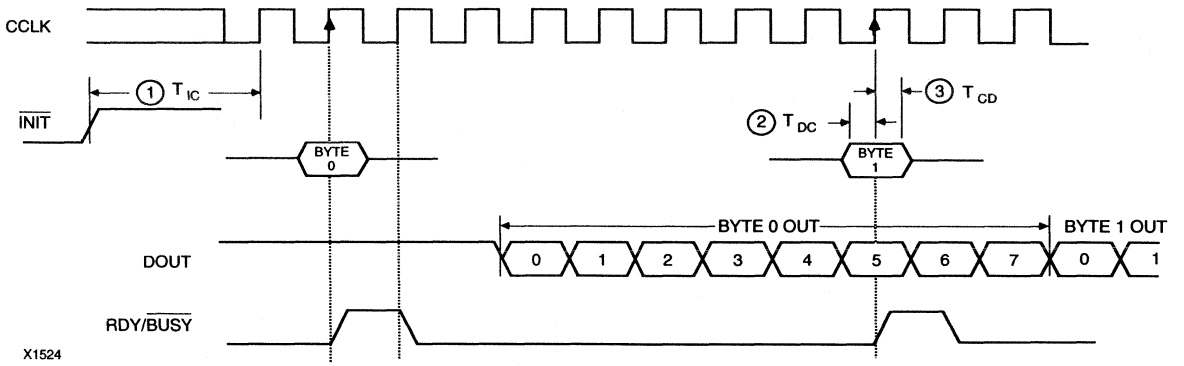
How to Delay Configuration After Power-Up

There are two methods to delay configuration after power-up: Put a logic Low on the $\overline{\text{PROGRAM}}$ input, or pull the bidirectional $\overline{\text{INIT}}$ pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27).

A Low on the $\overline{\text{PROGRAM}}$ input is the more radical approach, and is recommended when the power-supply rise time is excessive or poorly defined. As long as $\overline{\text{PROGRAM}}$ is Low, the XC4000 device keeps clearing its configuration memory. When $\overline{\text{PROGRAM}}$ goes High, the configuration memory is cleared one more time, followed by the beginning of configuration, provided the INIT input is not externally held Low. Note that a Low on the $\overline{\text{PROGRAM}}$ input automatically forces a Low on the $\overline{\text{INIT}}$ output.

Using an open-collector or open-drain driver to hold $\overline{\text{INIT}}$ Low before the beginning of configuration, causes the LCA device to wait after having completed the configuration memory clear operation. When $\overline{\text{INIT}}$ is no longer held Low externally, the device determines its configuration mode by capturing its status inputs, and is ready to start the configuration process. A master device waits an additional max 250 µs to make sure that all slaves in the potential daisy-chain have seen $\overline{\text{INIT}}$ being High.

Synchronous Peripheral Mode Programming Switching Characteristics



	Description	Symbol		Min	Max	Units
CCLK	$\overline{INIT}$ (High) Setup time required	1	T_{IC}	5		µs
	DIN Setup time required	2	T_{DC}	60		ns
	DIN Hold time required	3	T_{CD}	0		ns
	CCLK High time		T_{CCH}	50		ns
	CCLK Low time		T_{CCL}	60		ns
	CCLK Frequency		F_{CC}		8	MHz

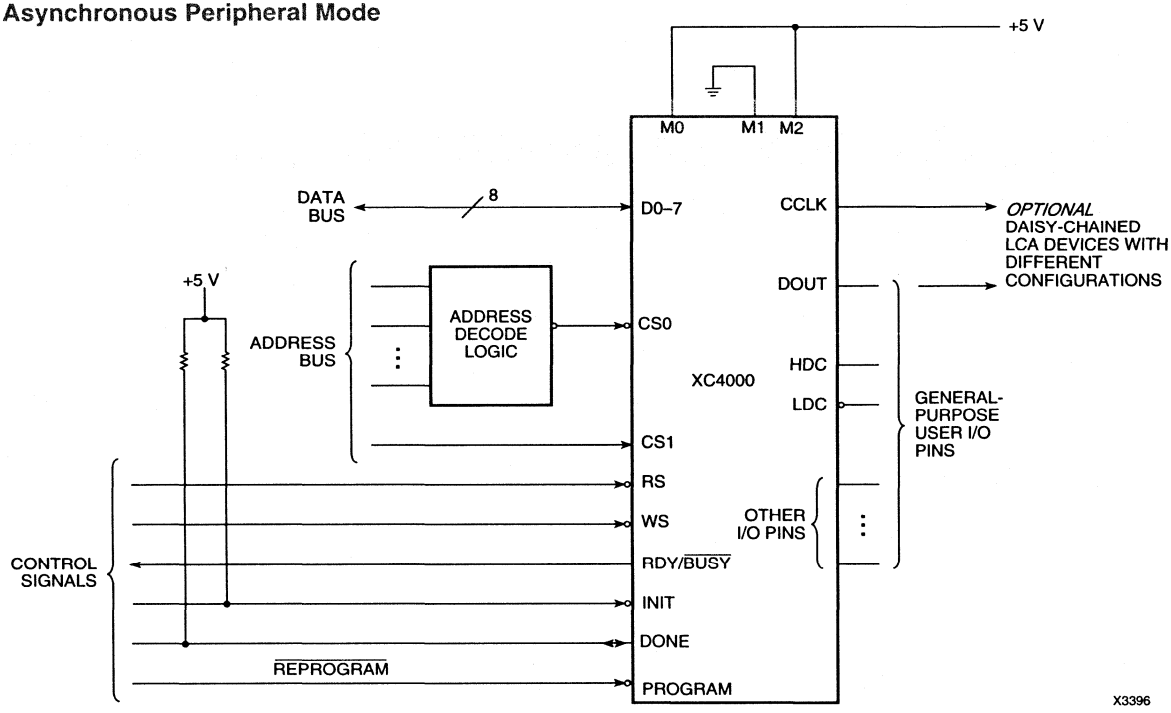
Notes: Peripheral Synchronous mode can be considered Slave Parallel mode. An external CCLK provides timing, clocking in the **first** data byte on the **second** rising edge of CCLK after $\overline{INIT}$ goes High. Subsequent data bytes are clocked in on every eighth consecutive rising edge of CCLK.

The RDY/$\overline{BUSY}$ line goes High for one CCLK period after data has been clocked in, although synchronous operation does not require such a response.

The pin name RDY/$\overline{BUSY}$ is a misnomer; in Synchronous Peripheral mode this is really an ACKNOWLEDGE signal.

Note that data starts to shift out serially on the DOUT pin 0.5 CLK periods after it was loaded in parallel. This obviously requires additional CCLK pulses after the last byte has been loaded.

Asynchronous Peripheral Mode



Write to LCA

Asynchronous Peripheral mode uses the trailing edge of the logic AND condition of the $\overline{CS0}$, CS1 and $\overline{WS}$ inputs to accept byte-wide data from a microprocessor bus. In the lead LCA device, this data is loaded into a double-buffered UART-like parallel-to-serial converter and is serially shifted into the internal logic. The lead LCA device presents the preamble data (and all data that overflows the lead device) on the DOUT pin.

The RDY/$\overline{BUSY}$ output from the lead LCA device acts as a handshake signal to the microprocessor. RDY/$\overline{BUSY}$ goes Low when a byte has been received, and goes High again when the byte-wide input buffer has transferred its information into the shift register, and the buffer is ready to receive new data. The length of the $\overline{BUSY}$ signal depends on the activity in the UART. If the shift register had been empty when the new byte was received, the $\overline{BUSY}$ signal lasts for only two CCLK periods. If the shift register was still full when the new byte was received, the $\overline{BUSY}$ signal can be as long as nine CCLK periods.

Note that after the last byte has been entered, only seven of its bits are shifted out. CCLK remains High with DOUT equal to bit 6 (the next-to-last bit) of the last byte entered.

Status Read

The logic AND condition of the $\overline{CS0}$, CS1and $\overline{RS}$ inputs puts the device status on the Data bus.

D7 = High indicates Ready
D7 - Low indicates Busy
D0 through D6 go unconditionally Low

It is mandatory that the whole start-up sequence be started and completed by one byte-wide input. Otherwise, the pins used as Write Strobe or Chip Enable might become active outputs and inteffere with the final byte transfer. If this transfer does not occur, the start-up sequence will not be completed all the way to the finish (point F in Figure 21 on page 2-29). At worst, the internal reset will not be released; at best, Readback and Boundary Scan will be inhibited. The length-count value, as generated by MAKEPROM, is supposed to ensure that these problems never occur.

Although RDY/$\overline{BUYSY}$ is brought out as a separate signal, microprocessors can more easily read this information on one of the data lines. For this purpose, D7 represents the RDY/$\overline{BUSY}$ status when $\overline{RS}$ is Low, $\overline{WS}$ is High, and the two chip select lines are both active.

How to Delay Configuration After Power-Up

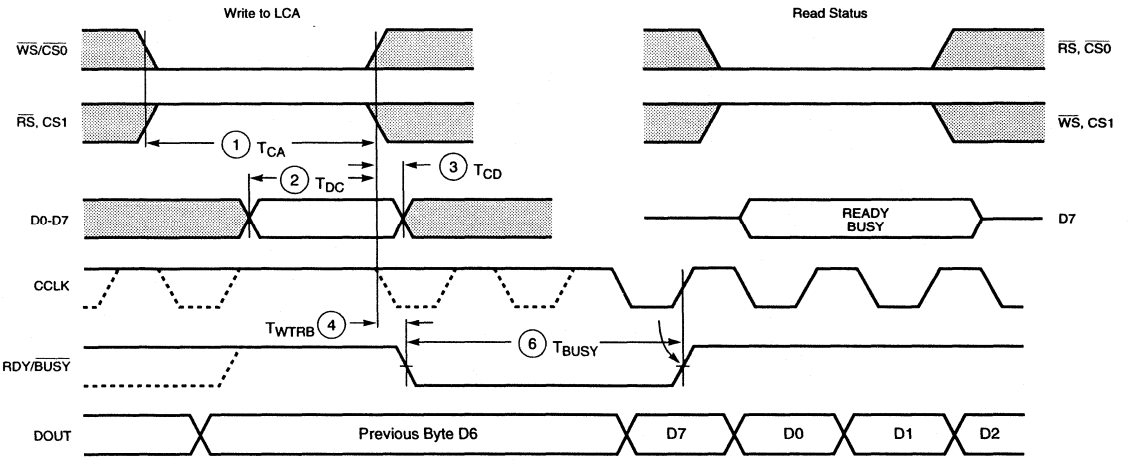
There are two methods to delay configuration after power-up: Put a logic Low on the $\overline{PROGRAM}$ input, or pull the bidirectional $\overline{INIT}$ pin Low, using an open-collector (open-drain) driver. (See also Figure 20 on page 2-27).

A Low on the $\overline{PROGRAM}$ input is the more radical approach, and is recommended when the power-supply rise time is excessive or poorly defined. As long as $\overline{PROGRAM}$

is Low, the XC4000 device keeps clearing its configuration memory. When $\overline{PROGRAM}$ goes High, the configuration memory is cleared one more time, followed by the beginning of configuration, provided the INIT input is not externally held Low. Note that a Low on the $\overline{PROGRAM}$ input automatically forces a Low on the $\overline{INIT}$ output.

Using an open-collector or open-drain driver to hold $\overline{INIT}$ Low before the beginning of configuration, causes the LCA device to wait after having completed the configuration memory clear operation. When $\overline{INIT}$ is no longer held Low externally, the device determines its configuration mode by capturing its status inputs, and is ready to start the configuration process. A master device waits an additional max 250 µs to make sure that all slaves in the potential daisy-chain have seen $\overline{INIT}$ being High.

Asynchronous Peripheral Mode Programming Switching Characteristics

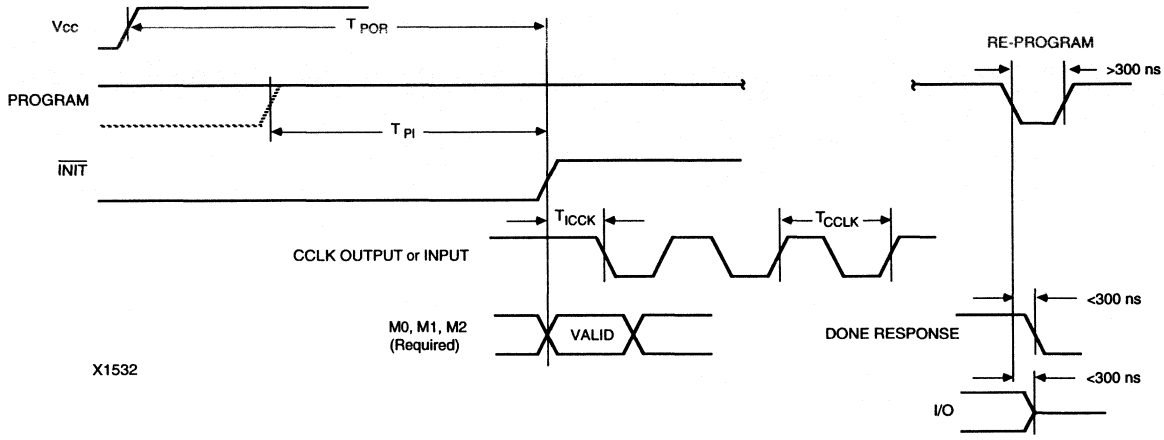


	Description	Symbol		Min	Max	Units
Write	Effective Write time required ($\overline{CS0}$, $\overline{WS}$ = Low, $\overline{RS}$, CS1 = High)	1	T_{CA}	100		ns
RDY	DIN Setup time required	2	T_{DC}	60		ns
	DIN Hold time required	3	T_{CD}	0		ns
	RDY/$\overline{BUSY}$ delay after end of $\overline{WS}$	4	T_{WTRB}		60	ns
	Earliest next $\overline{WS}$ after end of $\overline{BUSY}$	5	T_{RBWT}	0		ns
	$\overline{BUSY}$ Low output (Note 4)	6	T_{BUSY}	2	9	CCLK Periods

Notes:
1. Configuration must be delayed until the $\overline{INIT}$ of all LCA devices is High.
2. Time from end of $\overline{WS}$ to CCLK cycle for the new byte of data depends on completion of previous byte processing and the phase of the internal timing generator for CCLK.
3. CCLK and DOUT timing is tested in slave mode.
4. T_{BUSY} indicates that the double-buffered parallel-to-serial converter is not yet ready to receive new data. The shortest T_{BUSY} occurs when a byte is loaded into an empty parallel-to-serial converter. The longest T_{BUSY} occurs when a new word is loaded into the input register before the second-level buffer has started shifting out data.

This timing diagram shows very relaxed requirements:
Data need not be held beyond the rising edge of $\overline{WS}$. $\overline{BUSY}$ will go active within 60 ns after the end of $\overline{WS}$.
$\overline{WS}$ may be asserted immediately after the end of $\overline{BUSY}$.

General LCA Switching Characteristics



Master Modes

	Symbol	Min	Max	Units
Power-On-Reset M0 = High	T_{POR}	10	40	ms
Power-On-Reset M0 = Low	T_{POR}	40	130	ms
Program Latency	T_{PI}	30	200	µs per CLB column
CCLK (output) Delay	T_{ICCK}	40	250	µs
CCLK (output) period (slow)	T_{CCLK}	640	2000	ns
CCLK (output) period (fast)	T_{CCLK}	100	250	ns

Slave and Peripheral Modes

	Symbol	Min	Max	Units
Power-On-Reset	T_{POR}	10	33	ms
Program Latency	T_{PI}	30	200	µs per CLB column
CCLK (input) Delay (required)	T_{ICCK}	4		µs
CCLK (input) period (required)	T_{CCLK}	125		ns

Note: At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms, otherwise delay configuration using $\overline{PROGRAM}$ until V_{CC} is valid.

Pin Functions During Configuration

CONFIGURATION MODE: <M2:M1:M0>						
SLAVE <1:1:1>	MASTER-SER <0:0:0>	SYN.PERIPH <0:1:1>	ASYN.PERIPH <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>	USER OPERATION
				A16	A16	PGI-I/O
				A17	A17	I/O
TDI	TDI	TDI	TDI	TDI	TDI	TDI-I/O
TCK	TCK	TCK	TCK	TCK	TCK	TCK-I/O
TMS	TMS	TMS	TMS	TMS	TMS	TMS-I/O
						SGI-I/O
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	(O)
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	(I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	(I)
						PGI-I/O
HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	I/O
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	I/O
$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	$\overline{\text{INIT-ERROR}}$	I/O
						SGI-I/O
DONE	DONE	DONE	DONE	DONE	DONE	DONE
$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$ (I)	$\overline{\text{PROGRAM}}$
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	DATA 7 (I)	I/O
						PGI-I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	DATA 6 (I)	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	DATA 5 (I)	I/O
			$\overline{\text{CS0}}$ (I)			I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)		I/O
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	DATA 3 (I)	I/O
			$\overline{\text{RS}}$ (I)			I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	DATA 2 (I)	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	DATA 1 (I)	I/O
		RDY/$\overline{\text{BUSY}}$	RDY/$\overline{\text{BUSY}}$	$\overline{\text{RCLK}}$	$\overline{\text{RCLK}}$	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	DOUT	SGI-I/O
CCLK (I)	CCLK (O)	CCLK (O)	CCLK (O)	CCLK(O)	CCLK (O)	CCLK (I)
TDO	TDO	TDO	TDO	TDO	TDO	TDO-(O)
			$\overline{\text{WS}}$ (I)	A0	A0	I/O
				A1	A1	PGI-I/O
			CS1 (I)	A2	A2	I/O
				A3	A3	I/O
				A4	A4	I/O
				A5	A5	I/O
				A6	A6	I/O
				A7	A7	I/O
				A8	A8	I/O
				A9	A9	I/O
				A10	A10	I/O
				A11	A11	I/O
				A12	A12	I/O
				A13	A13	I/O
				A14	A14	I/O
				A15	A15	SGI-I/O
						ALL OTHERS

Represents a 50 kΩ to 100 kΩ pull-up before and during configuration

* INIT is an open-drain output during configuration

(I) Represents an input

X5265

Before and during configuration, all outputs that are not used for the configuration process are 3-stated with a 50 kΩ to 100 kΩ pull-up resistor.

Pin Descriptions

Permanently Dedicated Pins

V_{CC}

Eight or more (depending on package type) connections to the nominal +5 V supply voltage. All must be connected.

GND

Eight or more (depending on package type) connections to ground. All must be connected.

CCLK

During configuration, Configuration Clock is an output of the LCA in Master modes or asynchronous Peripheral mode, but is an input to the LCA in Slave mode and Synchronous Peripheral mode.

After configuration, CCLK has a weak pull-up resistor and can be selected as Readback Clock.

DONE

This is a bidirectional signal with optional pull-up resistor.

As an output, it indicates the completion of the configuration process. The configuration program determines the exact timing, the clock source for the Low-to-High transition, and enable of the pull-up resistor.

As an input, a Low level on DONE can be configured to delay the global logic initialization or the enabling of outputs

$\overline{\text{PROGRAM}}$

This is an active Low input that forces the LCA to clear its configuration memory.

When $\overline{\text{PROGRAM}}$ goes High, the LCA finishes the current clear cycle and executes another complete clear cycle, before it goes into a WAIT state and releases $\overline{\text{INIT}}$.

User I/O Pins that can have Special Functions

RDY/$\overline{\text{BUSY}}$

During peripheral modes, this pin indicates when it is appropriate to write another byte of data into the LCA device. The same status is also available on D7 in asynchronous peripheral mode, if a read operation is performed when the device is selected. After configuration, this is a user-programmable I/O pin.

$\overline{\text{RCLK}}$

During Master Parallel configuration, each change on the A0-15 outputs is preceded by a rising edge on $\overline{\text{RCLK}}$, a redundant output signal. After configuration, this is a user-programmable I/O pin.

M0, M1, M2

As Mode inputs, these pins are sampled before the start of configuration to determine the configuration mode to be used.

After configuration, M0 and M2 can be used as inputs, and M1 can be used as a 3-state output. These three pins have no associated input or output registers.

These pins can be user inputs or outputs only when called out by special schematic definitions.

TDO

If boundary scan is used, this is the Test Data Output.

If boundary scan is not used, this pin is a 3-state output without a register, after configuration is completed.

This pin can be user output only when called out by special schematic definitions.

TDI,TCK, TMS

If boundary scan is used, these pins are Test Data In, Test Clock, and Test Mode Select inputs respectively coming directly from the pads, bypassing theIOBs. These pins can also be used as inputs to the CLB logic after configuration is completed.

If the boundary scan option is not selected, all boundary scan functions are inhibited once configuration is completed, and these pins become user-programmable I/O.

Note:
The XC4000 families have no Powerdown control input; use the global 3-state net instead.
The XC4000 families have no dedicated Reset input. Any user I/O can be configured to drive the global Set/Reset net.

HDC

High During Configuration is driven High until configuration is completed. It is available as a control output indicating that configuration is not yet completed. After configuration, this is a user-programmable I/O pin.

$\overline{LDC}$

Low During Configuration is driven Low until configuration. It is available as a control output indicating that configuration is not yet completed. After configuration, this is a user-programmable I/O pin.

$\overline{INIT}$

Before and during configuration, this is a bidirectional signal. An external pull-up resistor is recommended.

As an active-Low open-drain output, $\overline{INIT}$ is held Low during the power stabilization and internal clearing of the configuration memory. As an active-Low input, it can be used to hold the LCA device in the internal WAIT state before the start of configuration. Master mode devices stay in a WAIT state an additional 30 to 300 µs after $\overline{INIT}$ has gone High.

During configuration, a Low on this output indicates that a configuration data error has occurred. After configuration, this is a user-programmable I/O pin.

PGCK1 - PGCK4

Four Primary Global Inputs each drive a dedicated internal global net with short delay and minimal skew. If not used for this purpose, any of these pins is a user-programmable I/O.

SGCK1 - SGCK4

Four Secondary Global Inputs can each drive a dedicated internal global net, that alternatively can also be driven from internal logic. If not used for this purpose, any of these pins is a user-programmable I/O pin.

$\overline{CS0}$, CS1, $\overline{WS}$, $\overline{RS}$

These four inputs are used in Peripheral mode. The chip is selected when $\overline{CS0}$ is Low and CS1 is High. While the chip is selected, a Low on Write Strobe ($\overline{WS}$) loads the data present on the D0 - D7 inputs into the internal data buffer; a Low on Read Strobe ($\overline{RS}$) changes D7 into a status output: High if Ready, Low if Busy, and D0...D6 are active Low. $\overline{WS}$ and $\overline{RS}$ should be mutually exclusive, but if both are Low simultaneously, the Write Strobe overrides. After configuration, these are user-programmable I/O pins.

A0 - A17

During Master Parallel mode, these 18 output pins address the configuration EPROM. After configuration, these are user-programmable I/O pins.

D0 - D7

During Master Parallel and Peripheral configuration modes, these eight input pins receive configuration data. After configuration, they are user-programmable I/O pins.

DIN

During Slave Serial or Master Serial configuration modes, this is the serial configuration data input receiving data on the rising edge of CCLK.

During parallel configuration modes, this is the D0 input. After configuration, DIN is a user-programmable I/O pin.

DOUT

During configuration in any mode, this is the serial configuration data output that can drive the DIN of daisy-chained slave LCA devices. DOUT data changes on the falling edge of CCLK, one-and-a-half CCLK periods after it was received at the DIN input. After configuration, DOUT is a user-programmable I/O pin.

Unrestricted User-Programmable I/O Pins

I/O

A pin that can be configured to be input and/or output after configuration is completed. Before configuration is completed, these pins have an internal high-value pull-up resistor that defines the logic level as High.

Before and during configuration, all outputs that are not used for the configuration process are 3-stated with a 50 kΩ to 100 kΩ pull-up resistor.

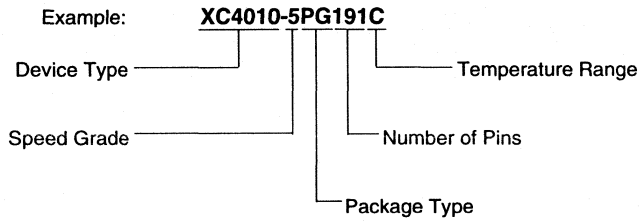
For a detailed description of the device architecture, see page 2-9 through 2-31.

For a detailed description of the configuration modes and their timing, see pages 2-32 through 2-55.

For detailed lists of package pinouts, see pages 2-57 through 2-67, 2-70, 2-81 through 2-85, and 2-100 through 2-101.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS		84	100			120	144	156	160	164	191	196	208		223	225	240		299
TYPE		PLAST. PLCC	PLAST. PQFP	PLAST. VQFP	TOP BRAZED CQFP	CERAM. PGA	PLAST. TQFP	CERAM PGA	PLAST. PQFP	TOP BRAZED CQFP	CERAM. PGA	TOP BRAZED CQFP	PLAST. PQFP	METAL PQFP	CERAM. PGA	CERAM. BGA	PLAST. PQFP	METAL PQFP	METAL PQFP
CODE		PC84	PQ100	VQ100	CB100	PG120	TQ144	PG156	PQ160	CB164	PG191	CB196	PQ208	MQ208	PG223	BG225	PQ240	MQ240	PG299
XC4003	-6	C I	C I			C I													
	-5	C	C			C													
	-4	C	C			C													
XC4005	-10							M B		M B									
	-6	C I						C I M B	C I	M B			C I						
	-5	C I						C I	C I				C I						
	-4	C						C	C				C						
XC4006	-6	C I						C I	C I				C I						
	-5	C I						C I	C I				C I						
	-4	C						C	C				C						
XC4008	-6	C I							C I		C I		(C I)	C I					
	-5	C I							C I		C I		(C I)	C I					
	-4	C							C		C		(C)	C					
XC40010	-10										M B	M B							
	-6	C I							C I		C I M B	M B	(C I)	C I		C I			
	-5	C I							C I		C I		(C I)	C I		C I			
	-4	C							C		C		(C)	C		C			
XC4010D	-6	C I							C I										
	-5	C I							C I										
	-4																		
XC4013	-6													C I	C I (M B)	C I	(C I)	C I	
	-5													C I	C I	C I	C I	C I	
	-4													C	C	C	(C)	C	
XC4025	-6													C I					C I
	-5													C					C
	-4																		
XC4002A	-6	C I	C I	C I		C I													
	-5	C	C	C		C													
	-4	C	C	C		C													
XC4003A	-10				M B	M B													
	-6	C I	C I	C I	M B	C I M B													
	-5	C	C	C		C													
	-4	C	C	C		C													
XC4004A	-6	C I				C I	C I		C I										
	-5	C				C	C		C										
	-4	C				C	C		C										
XC4005A	-6	C I					C I	C I	C I				C I						
	-5	C					C	C	C				C						
	-4	C					C	C	C				C						
XC4003H	-6										C I		C I						
	-5										C		C						
	-4										C		C						
XC4005H	-6														C I		(C I)	C I	
	-5														C		(C)	C	
	-4														C		(C)	C	

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C
B = MIL-STD-883C Class B Parentheses indicate future product plans

XC4000
Logic Cell Array Family

Product Specifications

Features

- Third Generation Field-Programmable Gate Arrays
  - Abundant flip-flops
  - Flexible function generators
  - On-chip ultra-fast RAM
  - Dedicated high-speed carry-propagation circuit
  - Wide edge decoders (four per edge)
  - Hierarchy of interconnect lines
  - Internal 3-state bus capability
  - Eight global low-skew clock or signal distribution network
- Flexible Array Architecture
  - Programmable logic blocks and I/O blocks
  - Programmable interconnects and wide decoders
- Sub-micron CMOS Process
  - High-speed logic and Interconnect
  - Low power consumption
- Systems-Oriented Features
  - IEEE 1149.1-compatible boundary-scan logic support
  - Programmable output slew rate (2 modes)
  - Programmable input pull-up or pull-down resistors
  - 12-mA sink current per output
  - 24-mA sink current per output pair
- Configured by Loading Binary File
  - Unlimited reprogrammability
  - Six programming modes
- XACT Development System runs on '386/'486-type PC, NEC PC, Apollo, Sun-4, and Hewlett-Packard 700 series
  - Interfaces to popular design environments like Viewlogic, Mentor Graphics and OrCAD
  - Fully automatic partitioning, placement and routing
  - Interactive design editor for design optimization
  - 288 macros, 34 hard macros, RAM/ROM compiler

Description

The XC4000 family of Field-Programmable Gate Arrays (FPGAs) provides the benefits of custom CMOS VLSI, while avoiding the initial cost, time delay, and inherent risk of a conventional masked gate array.

The XC4000 family provides a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), interconnected by a powerful hierarchy of versatile routing resources, and surrounded by a perimeter of programmable Input/Output Blocks (IOBs).

XC4000 devices have generous routing resources to accommodate the most complex interconnect patterns. They are customized by loading configuration data into the internal memory cells. The FPGA can either actively read its configuration data out of external serial or byte-parallel PROM (master modes), or the configuration data can be written into the FPGA (slave and peripheral modes).

The XC4000 family is supported by powerful and sophisticated software, covering every aspect of design: from schematic entry, to simulation, to automatic block placement and routing of interconnects, and finally the creation of the configuration bit stream.

Since Xilinx FPGAs can be reprogrammed an unlimited number of times, they can be used in innovative designs where hardware is changed dynamically, or where hardware must be adapted to different user applications. FPGAs are ideal for shortening the design and development cycle, but they also offer a cost-effective solution for production rates well beyond 1000 systems per month.

For a detailed description of the device features, architecture, configuration methods and pin descriptions, see pages 2-9 through 2-45.

Table 1. The XC4000 Family of Field-Programmable Gate Arrays

Device	XC4003	XC4005	XC4006	XC4008	XC4010/10D	XC4013	XC4025
Appr. Gate Count	3,000	5,000	6,000	8,000	10,000	13,000	25,000
CLB Matrix	10 x 10	14 x 14	16 x 16	18 x 18	20 x 20	24 x 24	32 x 32
Number of CLBs	100	196	256	324	400	576	1,024
Number of Flip-Flops	360	616	768	936	1,120	1,536	2,560
Max Decode Inputs (per side)	30	42	48	54	60	72	96
Max RAM Bits	3,200	6,272	8,192	10,368	12,800*	18,432	32,768
Number of IOBs	80	112	128	144	160	192	256

*XC4010D has no RAM

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to 7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to 7	V
V_{TS}	Voltage applied to 3-state output	–0.5 to 7	V
T_{STG}	Storage temperature (ambient)	–65 to + 150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in. = 1.5 mm)	+ 260	°C
T_J	Junction temperature	+ 150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description			Min	Max	Units
V_{CC}	Supply voltage relative to GND	Commercial	0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND	Industrial	–40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND	Military	–55°C to 125°C	4.5	5.5	V
V_{IH}	High-level input voltage (XC4000 has TTL-like input thresholds)			2.0	V_{CC}	V
V_{IL}	Low-level input voltage (XC4000 has TTL-like input thresholds)			0	0.8	V
T_{IN}	Input signal transition time				250	ns

DC Characteristics Over Operating Conditions

Symbol	Description	Min	Max	Units
V_{OH}	High-level output voltage @ I_{OH} = –4.0 mA, V_{CC} min	2.4		V
V_{OL}	Low-level output voltage @ I_{OL} = 12.0 mA, V_{CC} max (Note 1)		0.4	V
I_{CCO}	Quiescent LCA supply current (Note 2)		10	mA
I_{IL}	Leakage current	–10	+10	µA
C_{IN}	Input capacitance (sample tested)		15	pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0V (sample tested)	0.02	0.25	mA
I_{RLL}	Horizontal Long Line pull-up (when selected) @ logic Low	0.2	2.5	mA

Note: 1. With 50% of the outputs simultaneously sinking 12 mA.
2. With no output current loads, no active input or longline pull-up resistors, all package pins at V_{CC} or GND, and the LCA configured with a MakeBits tie option.

Wide Decoder Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
Full length, both pull-ups, inputs from IOB I-pins	T_{WAF}	XC4003	9.0	8.0	5.0	ns
		XC4005	10.0	9.0	6.0	ns
		XC4006	11.0	10.0	7.0	ns
		XC4008	12.0	11.0	8.0	ns
		XC4010	13.0	12.0	9.0	ns
		XC4013	15.0	14.0	11.0	ns
Full length, both pull-ups inputs from internal logic	T_{WAFL}	XC4003	12.0	11.0	7.0	ns
		XC4005	13.0	12.0	8.0	ns
		XC4006	14.0	13.0	9.0	ns
		XC4008	15.0	14.0	10.0	ns
		XC4010	16.0	15.0	11.0	ns
		XC4013	18.0	17.0	13.0	ns
Half length, one pull-up inputs from IOB I-pins	T_{WAO}	XC4003	9.0	8.0	6.0	ns
		XC4005	10.0	9.0	7.0	ns
		XC4006	11.0	10.0	8.0	ns
		XC4008	12.0	11.0	9.0	ns
		XC4010	13.0	12.0	10.0	ns
		XC4013	15.0	14.0	12.0	ns
Half length, one pull-up inputs from internal logic	T_{WAOL}	XC4003	12.0	11.0	8.0	ns
		XC4005	13.0	12.0	9.0	ns
		XC4006	14.0	13.0	10.0	ns
		XC4008	15.0	14.0	11.0	ns
		XC4010	16.0	15.0	12.0	ns
		XC4013	18.0	17.0	14.0	ns

Note: These delays are specified from the decoder input to the decoder output. For pin-to-pin delays, add the input delay (T_{PID}) and output delay (T_{OPF} or T_{OPS}), as listed on page 2-52.

Global Buffer Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

		Speed Grade	-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
Global Signal Distribution						
From pad through **primary** buffer, to any clock K	T_{PG}	XC4003	7.8	5.8	5.1	ns
		XC4005	8.0	6.0	5.5	ns
		XC4006	8.2	6.2	5.7	ns
		XC4008	8.6	6.6	6.1	ns
		XC4010	9.0	7.0	6.5	ns
		XC4013	10.0	8.0	7.5	ns
From pad through **secondary** buffer, to any clock K	T_{SG}	XC4003	8.8	6.8	6.3	ns
		XC4005	9.0	7.0	6.7	ns
		XC4006	9.2	7.2	6.9	ns
		XC4008	9.6	7.6	7.3	ns
		XC4010	10.0	8.0	7.7	ns
		XC4013	11.0	9.0	8.7	ns

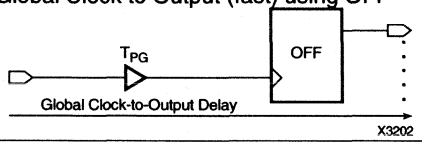
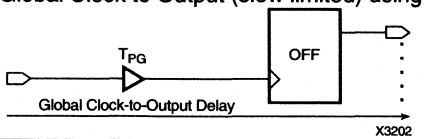
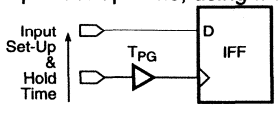
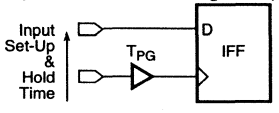
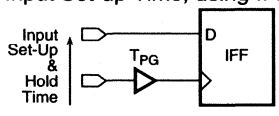
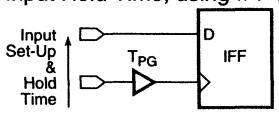
Horizontal Longline Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

		Speed Grade	-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
TBUF driving a Horizontal Longline (L.L.) I going High or Low to L.L. going High or Low, while T is Low, i.e. buffer is constantly active	T_{IO1}	XC4003	8.8	6.2	4.4	ns
		XC4005	10.0	7.0	5.5	ns
		XC4006	10.6	7.5	6.0	ns
		XC4008	11.1	8.0	6.5	ns
		XC4010	11.7	8.5	7.0	ns
		XC4013	13.0	9.5	7.5	ns
I going Low to L.L. going from resistive pull-up High to active Low, (TBUF configured as open drain)	T_{IO2}	XC4003	9.3	6.7	5.0	ns
		XC4005	10.5	7.5	6.0	ns
		XC4006	11.1	8.0	6.5	ns
		XC4008	11.6	8.5	7.0	ns
		XC4010	12.2	9.0	7.5	ns
		XC4013	13.5	10.0	8.0	ns
T going Low to L.L. going from resistive pull-up or floating High to active Low, (TBUF configured as open drain or active buffer with I = Low)	T_{ON}	XC4003	10.7	9.0	7.2	ns
		XC4005	12.0	10.0	8.0	ns
		XC4006	12.6	10.5	8.5	ns
		XC4008	13.2	11.0	9.0	ns
		XC4010	13.8	11.5	9.5	ns
		XC4013	15.1	12.6	11.1	ns
T going High to TBUF going inactive, not driving L.L.	T_{OFF}	All devices	3.0	2.0	1.8	ns
T going High to L.L. going from Low to High, pulled up by a single resistor	T_{PUS}	XC4003	24.0	26.0	14.0	ns
		XC4005	26.0	22.0	16.0	ns
		XC4006	28.0	24.0	18.0	ns
		XC4008	30.0	26.0	20.0	ns
		XC4010	32.0	28.0	22.0	ns
		XC4013	36.0	32.0	26.0	ns
T going High to L.L. going from Low to High, pulled up by two resistors	T_{PUF}	XC4003	11.6	9.0	7.0	ns
		XC4005	12.0	10.0	8.0	ns
		XC4006	13.0	11.0	9.0	ns
		XC4008	14.0	12.0	10.0	ns
		XC4010	15.0	13.0	11.0	ns
		XC4013	17.0	15.0	13.0	ns

Guaranteed Input and Output Parameters (Pin-to-Pin)

All values listed below are tested directly, and guaranteed over the operating conditions. The same parameters can also be derived indirectly from the IOB and Global Buffer specifications. The XACT delay calculator uses this indirect method. When there is a discrepancy between these two methods, the values listed below should be used, and the derived values must be ignored.

Description	Symbol	Speed Grade: Device	-6	-5	-4	Units
Global Clock to Output (fast) using OFF 	T_{ICKOF} (Max)	XC4003	15.1	12.5	11.6	ns
		XC4005	15.5	13.0	12.0	ns
		XC4006	15.7	13.2	12.2	ns
		XC4008	16.1	13.6	12.6	ns
		XC4010	16.5	14.0	13.0	ns
		XC4013	17.5	15.0	14.0	ns
Global Clock to Output (slew limited) using OFF 	T_{ICKO} (Max)	XC4003	19.9	15.2	14.4	ns
		XC4005	20.5	16.0	15.0	ns
		XC4006	20.7	16.2	15.2	ns
		XC4008	21.1	16.6	15.6	ns
		XC4010	21.5	17.0	16.0	ns
		XC4013	22.5	18.0	17.0	ns
Input Set-up Time, using IFF (fast) 	T_{PSUF} (Min)	XC4003	2.4	2.0	1.6	ns
		XC4005	2.0	1.5	1.2	ns
		XC4006	1.8	1.3	1.0	ns
		XC4008	1.4	0.9	0.6	ns
		XC4010	1.0	0.5	0.2	ns
		XC4013	0.5	0	0	ns
Input Hold time, using IFF (fast) 	T_{PHF} (Min)	XC4003	5.1	4.0	4.0	ns
		XC4005	5.5	4.5	4.5	ns
		XC4006	5.7	4.7	4.7	ns
		XC4008	6.1	5.1	5.1	ns
		XC4010	6.5	5.5	5.5	ns
		XC4013	7.5	6.5	6.5	ns
Input Set-up Time, using IFF (with delay) 	T_{PSU} (Min)	XC4003	21.5	18.5	12.0	ns
		XC4005	21.0	18.0	12.0	ns
		XC4006	20.8	17.8	12.0	ns
		XC4008	20.4	17.4	12.0	ns
		XC4010	20.0	17.0	12.0	ns
		XC4013	19.0	16.0	12.0	ns
Input Hold Time, using IFF (with delay) 	T_{PH} (Min)	XC4003	0	0	0	ns
		XC4005	0	0	0	ns
		XC4006	0	0	0	ns
		XC4008	0	0	0	ns
		XC4010	0	0	0	ns
		XC4013	0	0	0	ns

Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture). When testing fast outputs, only one output switches. When testing slew-rate limited outputs, half the number of outputs on one side of the device are switching. These parameter values are tested and guaranteed for worst-case conditions of supply voltage and temperature, and also with the most unfavorable clock polarity choice.

T_{PDLI} for -4 Speed Grade

Pad to I1, I2 via transparent latch, with delay	XC4003	17.6 ns
	XC4005	17.9 ns
	XC4006	18.0 ns
	XC4008	18.3 ns
PRELIMINARY	XC4010	18.6 ns
	XC4013	19.3 ns

T_{PICKD} for -4 Speed Grade

Input set-up time pad to clock (IK) with delay	XC4003	15.6 ns
	XC4005	15.9 ns
	XC4006	16.0 ns
	XC4008	16.3 ns
PRELIMINARY	XC4010	16.6 ns
	XC4013	17.3 ns

X5282

IOB Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade	-6		-5		-4		
Description	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Input								
Propagation Delays								
Pad to I1, I2	T_{PID}		4.0		3.0		2.8	ns
Pad to I1, I2, via transparent latch (fast)	T_{PLI}		8.0		7.0		6.0	ns
Pad to I1, I2, via transparent latch (with delay)	T_{PDLI}		26.0		24.0		**	ns
Clock (IK) tol1, I2, (flip-flop)	T_{IKRI}		8.0		7.0		6.0	ns
Clock (IK) to I1, I2 (latch enable, active Low)	T_{IKLI}		8.0		7.0		6.0	ns
Set-up Time (Note 3)								
Pad to Clock (IK), fast	T_{PICK}	7.0		6.0		4.0		ns
Pad to Clock (IK) with delay	T_{PICKD}	25.0		24.0		**		ns
Hold Time (Note 3)								
Pad to Clock (IK), fast	T_{IKPI}	1.0		1.0		1.0		ns
Pad to Clock (IK) with delay	T_{IKPID}	neg		neg		neg		ns
Output								
Propagation Delays								
Clock (OK) to Pad (fast)	T_{OKPOF}		7.5		7.0		6.5	ns
same (slew rate limited)	T_{OKPOS}		11.5		10.0		9.5	ns
Output (O) to Pad (fast)	T_{OPF}		9.0		7.0		5.5	ns
same (slew-rate limited)	T_{OPS}		13.0		10.0		8.5	ns
3-state to Pad begin hi-Z (slew-rate independent)	T_{TSHZ}		9.0		7.0		6.5	ns
3-state to Pad active and valid (fast)	T_{TSONF}		13.0		10.0		9.5	ns
same (slew -rate limited)	T_{TSONS}		17.0		13.0		12.5	ns
Set-up and Hold Times								
Output (O) to clock (OK) set-up time	T_{OOK}	8.0		6.0		5.5		ns
Output (O) to clock (OK) hold time	T_{OKO}	0		0		0		ns
Clock								
Clock High or Low time	T_{CH}/T_{CL}	5.0		4.0		4.0		ns
Global Set/Reset								
Delay from GSR net through Q to I1, I2	T_{RRI}		14.5		13.5		13.5	ns
Delay from GSR net to Pad	T_{RPO}		18.0		17.0		14.0	ns
GSR width*	T_{MRW}	21.0		18.0		18.0		ns

* Timing is based on the XC4005. For other devices see XACT timing calculator.
** See preceding page

Notes: 1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture). **Slew rate limited** output rise/fall times are approximately two times longer than **fast** output rise/fall times. **A maximum total external capacitive load for simultaneous fast mode switching in the same direction is 200 pF per power/ground pin pair. For slew-rate limited outputs this total is two times larger. Exceeding this maximum capacitive load can result in ground bounce of >1.5 V amplitude, <5 ns duration, which might cause problems when the LCA drives clocks and other asynchronous signals.**

2. Voltage levels of unused (bonded and unbonded) pads must be valid logic levels. Each can be configured with the internal pull-up or pull-down resistor or alternatively configured as a driven output or be driven from an external source.

3. Input pad setup times and hold times are specified with respect to the internal clock (IK). To calculate system setup time, subtract clock delay (clock pad to IK) from the specified input pad setup time value, but do not subtract below zero. Negative hold time means that the delay in the input data is adequate for the **external system hold time** to be zero, provided the input clock uses the Global signal distribution from pad to IK.

CLB Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade	-6		-5		-4		
Description	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Combinatorial Delays								
F/G inputs to X/Y outputs	T_{ILO}		6.0		4.5		4.0	ns
F/G inputs via H' to X/Y outputs	T_{IHO}		8.0		7.0		6.0	ns
C inputs via H' to X/Y outputs	T_{HHO}		7.0		5.0		4.5	ns
CLB Fast Carry Logic								
Operand inputs (F1,F2,G1,G4) to COUT	T_{OPCY}		7.0		5.5		5.0	ns
Add/Subtract input (F3) to COUT	T_{ASCY}		8.0		6.0		5.5	ns
Initialization inputs (F1,F3) to COUT	T_{INCY}		6.0		4.0		3.5	ns
C_{IN} through function generators to X/Y outputs	T_{SUM}		8.0		6.0		5.5	ns
C_{IN} to C_{OUT}, bypass function generators.	T_{BYP}		2.0		1.5		1.5	ns
Sequential Delays								
Clock K to outputs Q	T_{CKO}		5.0		3.0		3.0	ns
Set-up Time before Clock K								
F/G inputs	T_{ICK}	6.0		4.5		4.5		ns
F/G inputs via H'	T_{IHCK}	8.0		6.0		6.0		ns
C inputs via H1	T_{HHCK}	7.0		5.0		5.0		ns
C inputs via DIN	T_{DICK}	4.0		3.0		3.0		ns
C inputs via EC	T_{ECCK}	7.0		4.0		3.0		ns
C inputs via S/R, going Low (inactive)	T_{RCK}	6.0		4.5		4.0		ns
C_{IN} input via F'/G'	T_{CCK}	8.0		6.0		5.5		ns
C_{IN} input via F'/G' and H'	T_{CHCK}	10.0		7.5		7.3		ns
Hold Time after Clock K								
F/G inputs	T_{CKI}	0		0		0		ns
F/G inputs via H'	T_{CKIH}	0		0		0		ns
C inputs via H1	T_{CKHH}	0		0		0		ns
C inputs via DIN	T_{CKDI}	0		0		0		ns
C inputs via EC	T_{CKEC}	0		0		0		ns
C inputs via S/R, going Low (inactive)	T_{CKR}	0		0		0		ns
Clock								
Clock High time	T_{CH}	5.0		4.5		4.5		ns
Clock Low time	T_{CL}	5.0		4.5		4.5		ns
Set/Reset Direct								
Width (High)	T_{RPW}	5.0		4.0		4.0		ns
Delay from C inputs via S/R, going High to Q	T_{RIO}		9.0		8.0		7.0	ns
Master Set/Reset*								
Width (High or Low)	T_{MRW}	21.0		18.0		18.0		ns
Delay from Global Set/Reset net to Q	T_{MRQ}		33.0		31.0		28.0	ns

* Timing is based on the XC4005. For other devices see XACT timing calculator.

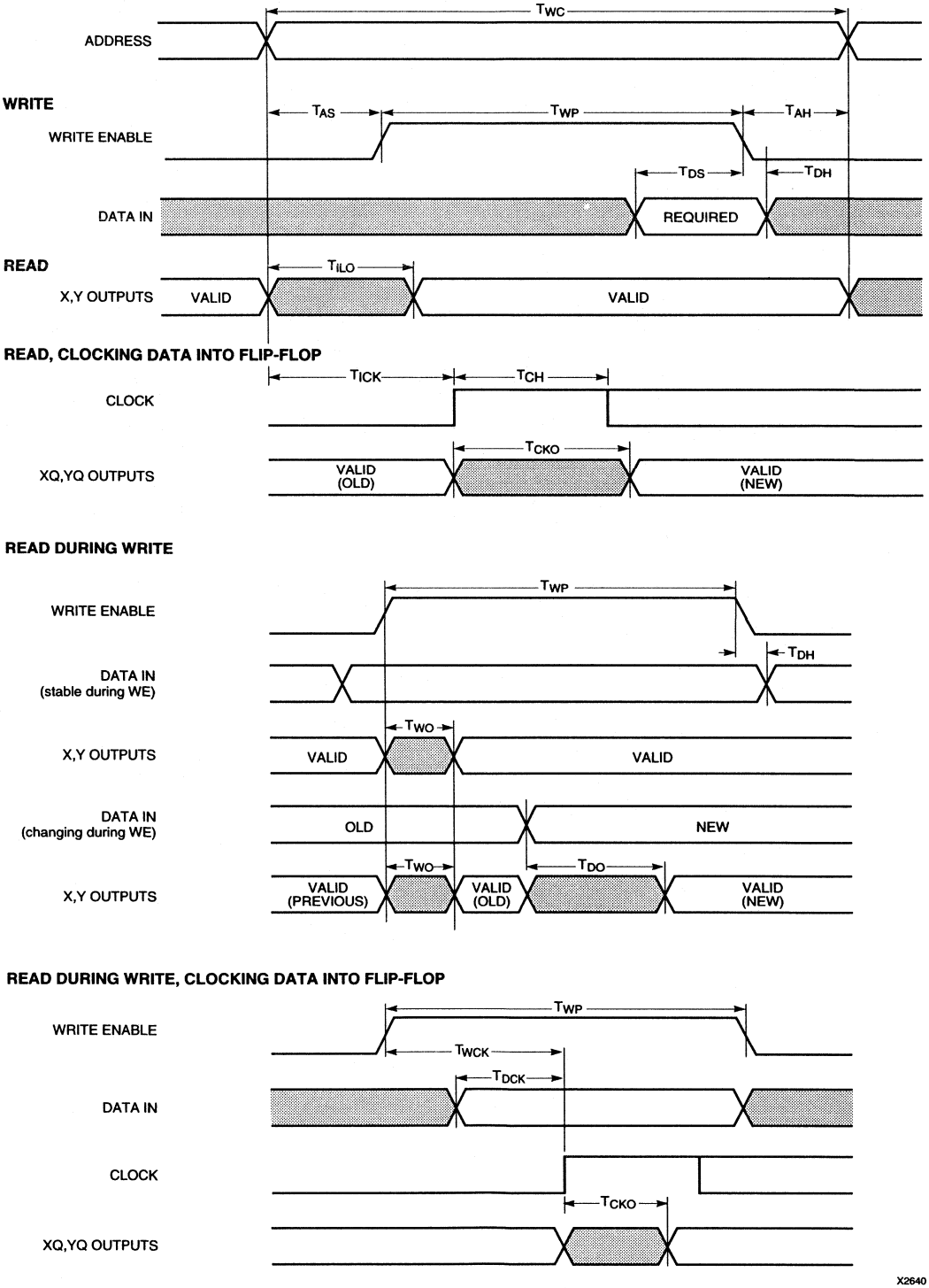
CLB Switching Characteristic Guidelines (continued)

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

CLB RAM Option	**Speed Grade**		**-6**		**-5**		**-4**		
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Write Operation									
Address write cycle time	16 x 2	T_{WC}	9.0		8.0		8.0		ns
	32 x 1	T_{WCT}	9.0		8.0		8.0		ns
Write Enable pulse width (High)	16 x 2	T_{WP}	5.0		4.0		4.0		ns
	32 x 1	T_{WPT}	5.0		4.0		4.0		ns
Address set-up time before beginning of WE	16 x 2	T_{AS}	2.0		2.0		2.0		ns
	32 x 1	T_{AST}	2.0		2.0		2.0		ns
Address hold time after end of WE	16 x 2	T_{AH}	2.0		2.0		2.0		ns
	32 x 1	T_{AHT}	2.0		2.0		2.0		ns
DIN set-up time before end of WE	16 x 2	T_{DS}	4.0		4.0		4.0		ns
	32 x 1	T_{DST}	5.0		5.0		5.0		ns
DIN hold time after end of WE	both	T_{DHT}	2.0		2.0		2.0		ns
Read Operation									
Address read cycle time	16 x 2	T_{RC}	7.0		5.5		5.0		ns
	32 x 1	T_{RCT}	10.0		7.5		7.0		ns
Data valid after address change	16 x 2	T_{ILO}		6.0		4.5		4.0	ns
(no Write Enable)	32 x 1	T_{IHO}		8.0		7.0		6.0	ns
Read Operation, Clocking Data into Flip-Flop									
Address setup time before clock K	16 x 2	T_{ICK}	6.0		4.5		4.5		ns
	32 x 1	T_{IHCK}	8.0		6.0		6.0		ns
Read During Write									
Data valid after WE going active	16 x 2	T_{WO}		12.0		10.0		9.0	ns
(DIN stable before WE)	32 x 1	T_{WOT}		15.0		12.0		11.0	ns
Data valid after DIN	16 x 2	T_{DO}		11.0		9.0		8.5	ns
(DIN change during WE)	32 x 1	T_{DOT}		14.0		11.0		11.0	ns
Read During Write, Clocking Data into Flip-Flop									
WE setup time before clock K	16 x 2	T_{WCK}	12.0		10.0		9.5		ns
	32 x 1	T_{WCKT}	15.0		12.0		11.5		ns
Data setup time before clock K	16 x 2	T_{DCK}	11.0		9.0		9.0		ns
	32 x 1	T_{DCKT}	14.0		11.0		11.0		ns

Note: Timing for the 16 x 1 RAM option is identical to 16 x 2 RAM timing

CLB RAM Timing Characteristics



XC4003 Pinouts

Pin Description	PC84	PQ100	PG120	Bound Scan
VCC	2	92	G3	–
I/O (A8)	3	93	G1	32
I/O (A9)	4	94	F1	35
I/O	–	95	E1	38
I/O	–	96	F2	41
I/O (A10)	5	97	F3	44
I/O (A11)	6	98	D1	47
–	–	–	E2*	–
I/O (A12)	7	99	C1	50
I/O (A13)	8	100	D2	53
–	–	–	E3*	–
–	–	–	B1*	–
I/O (A14)	9	1	C2	56
SGCK1 (A15,I/O)	10	2	D3	59
VCC	11	3	C3	–
GND	12	4	C4	–
PGCK1 (A16, I/O)	13	5	B2	62
I/O (A17)	14	6	B3	65
–	–	–	A1*	–
–	–	–	A2*	–
I/O (TDI)	15	7	C5	68
I/O (TCK)	16	8	B4	71
–	–	–	A3*	–
I/O (TMS)	17	9	B5	74
I/O	18	10	A4	77
I/O	–	–	C6	80
I/O	–	11	A5	83
I/O	19	12	B6	86
I/O	20	13	A6	89
GND	21	14	B7	–
VCC	22	15	C7	–
I/O	23	16	A7	92
I/O	24	17	A8	95
I/O	–	18	A9	98
I/O	–	–	B8	101
I/O	25	19	C8	104
I/O	26	20	A10	107
I/O	27	21	B9	110
I/O	–	22	A11	113
–	–	–	B10*	–
I/O	28	23	C9	116
SGCK2 (I/O)	29	24	A12	119
O (M1)	30	25	B11	122
GND	31	26	C10	–
I (M0)	32	27	C11	125†
VCC	33	28	D11	–
I (M2)	34	29	B12	126†
PGCK2 (I/O)	35	30	C12	127
I/O (HDC)	36	31	A13	130
–	–	–	B13*	–
–	–	–	E11*	–
I/O	–	32	D12	133
I/O ($\overline{LDC}$)	37	33	C13	136
I/O	38	34	E12	139
I/O	39	35	D13	142
I/O	–	36	F11	145
I/O	–	37	E13	148
I/O	40	38	F12	151
I/O ($\overline{ERR}$, $\overline{INIT}$)	41	39	F13	154
VCC	42	40	G12	–

Pin Description	PC84	PQ100	PG120	Bound Scan
GND	43	41	G11	–
I/O	44	42	G13	157
I/O	45	43	H13	160
I/O	–	44	J13	163
I/O	–	45	H12	166
I/O	46	46	H11	169
I/O	47	47	K13	172
I/O	48	48	J12	175
I/O	49	49	L13	178
–	–	–	K12*	–
–	–	–	J11*	–
I/O	50	50	M13	181
SGCK3 (I/O)	51	51	L12	184
GND	52	52	K11	–
DONE	53	53	L11	–
VCC	54	54	L10	–
$\overline{PROG}$	55	55	M12	–
I/O (D7)	56	56	M11	187
PGCK3 (I/O)	57	57	N13	190
–	–	–	N12*	–
–	–	–	L9*	–
I/O (D6)	58	58	M10	193
I/O	–	59	N11	196
I/O (D5)	59	60	M9	199
I/O ($\overline{CS0}$)	60	61	N10	202
I/O	–	62	L8	205
I/O	–	63	N9	208
I/O (D4)	61	64	M8	211
I/O	62	65	N8	214
VCC	63	66	M7	–
GND	64	67	L7	–
I/O (D3)	65	68	N7	217
I/O ($\overline{RS}$)	66	69	N6	220
I/O	–	70	N5	223
I/O	–	–	M6	226
I/O (D2)	67	71	L6	229
I/O	68	72	N4	232
I/O (D1)	69	73	M5	235
I/O (RCLK-$\overline{BUSY}$/RDY)	70	74	N3	238
–	–	–	M4*	–
–	–	–	L5*	–
I/O (D0, DIN)	71	75	N2	241
SGCK4 (DOUT, I/O)	72	76	M3	244
CCLK	73	77	L4	–
VCC	74	78	L3	–
O (TDO)	75	79	M2	–
GND	76	80	K3	–
I/O (A0, $\overline{WS}$)	77	81	L2	2
PGCK4 (A1, I/O)	78	82	N1	5
–	–	–	M1*	–
–	–	–	J3*	–
I/O (CS1, A2)	79	83	K2	8
I/O (A3)	80	84	L1	11
I/O (A4)	81	85	J2	14
I/O (A5)	82	86	K1	17
I/O	–	87	H3	20
I/O	–	88	J1	23
I/O (A6)	83	89	H2	26
I/O (A7)	84	90	H1	29
GND	1	91	G2	–

* Indicates unconnected package pins.
† Contributes only one bit (.i) to the boundary scan register.
Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 247 = BSCANT.UPD

XC4005 Pinouts

Pin Description	PC84	PQ160	PQ208	PG156	Bound Scan
VCC	2	142	183	H3	–
I/O (A8)	3	143	184	H1	44
I/O (A9)	4	144	185	G1	47
I/O	–	145	186	G2	50
I/O	–	146	187	G3	53
–	–	–	188*	–	–
–	–	–	189*	–	–
I/O (A10)	5	147	190	F1	56
I/O (A11)	6	148	191	F2	59
I/O	–	149	192	E1	62
I/O	–	150	193	E2	65
GND	–	151	194	F3	–
–	–	–	195*	–	–
–	–	–	196*	–	–
–	–	152*	197*	D1*	–
–	–	153*	198*	D2*	–
I/O (A12)	7	154	199	E3	68
I/O (A13)	8	155	200	C1	71
–	–	–	–	–	–
I/O	–	156	201	C2	74
I/O	–	157	202	D3	77
I/O (A14)	9	158	203	B1	80
SGCK1 (A15, I/O)	10	159	204	B2	83
VCC	11	160	205	C3	–
–	–	–	206*	–	–
–	–	–	207*	–	–
–	–	–	208*	–	–
–	–	–	1*	–	–
GND	12	1	2	C4	–
–	–	–	3*	–	–
PGCK1 (A16, I/O)	13	2	4	B3	86
I/O (A17)	14	3	5	A1	89
I/O	–	4	6	A2	92
I/O	–	5	7	C5	95
–	–	–	–	–	–
I/O (TDI)	15	6	8	B4	98
I/O (TCK)	16	7	9	A3	101
–	–	8*	10*	A4*	–
–	–	9*	11*	–	–
–	–	–	12*	–	–
–	–	–	13*	–	–
GND	–	10	14	C6	–
I/O	–	11	15	B5	104
I/O	–	12	16	B6	107
I/O (TMS)	17	13	17	A5	110
I/O	18	14	18	C7	113
–	–	–	19*	–	–
–	–	–	20*	–	–
I/O	–	15	21	B7	116
I/O	–	16	22	A6	119
I/O	19	17	23	A7	122
I/O	20	18	24	A8	125
GND	21	19	25	C8	–
VCC	22	20	26	B8	–
I/O	23	21	27	C9	128
I/O	24	22	28	B9	131
I/O	–	23	29	A9	134
I/O	–	24	30	B10	137
–	–	–	31*	–	–
–	–	–	32*	–	–
I/O	25	25	33	C10	140
I/O	26	26	34	A10	143
I/O	–	27	35	A11	146
I/O	–	28	36	B11	149
GND	–	29	37	C11	–
–	–	–	38*	–	–
–	–	–	39*	–	–
–	–	30*	40*	A12*	–
–	–	31*	41*	–	–
I/O	27	32	42	B12	152
I/O	–	33	43	A13	155
I/O	–	34	44	A14	158

Pin Description	PC84	PQ160	PQ208	PG156	Bound Scan
I/O	–	35	45	C12	161
–	–	–	–	–	–
I/O	28	36	46	B13	164
SGCK2 (I/O)	29	37	47	B14	167
O (M1)	30	38	48	A15	170
GND	31	39	49	C13	–
I (M0)	32	40	50	A16	173†
–	–	–	51*	–	–
–	–	–	52*	–	–
–	–	–	53*	–	–
–	–	–	54*	–	–
VCC	33	41	55	C14	–
I (M2)	34	42	56	B15	174†
PGCK2 (I/O)	35	43	57	B16	175
I/O (HDC)	36	44	58	D14	178
I/O	–	45	59	C15	181
–	–	–	–	–	–
I/O	–	46	60	D15	184
I/O	–	47	61	E14	187
I/O ($\overline{LDC}$)	37	48	62	C16	190
–	–	49*	63*	E15*	–
–	–	50*	64*	D16*	–
–	–	–	65*	–	–
–	–	–	66*	–	–
GND	–	51	67	F14	–
I/O	–	52	68	F15	193
I/O	–	53	69	E16	196
I/O	38	54	70	F16	199
I/O	39	55	71	G14	202
–	–	–	72*	–	–
–	–	–	73*	–	–
I/O	–	56	74	G15	205
I/O	–	57	75	G16	208
I/O	40	58	76	H16	211
I/O ($\overline{ERR}$, $\overline{INIT}$)	41	59	77	H15	214
VCC	42	60	78	H14	–
GND	43	61	79	J14	–
I/O	44	62	80	J15	217
I/O	45	63	81	J16	220
I/O	–	64	82	K16	223
I/O	–	65	83	K15	226
–	–	–	84*	–	–
–	–	–	85*	–	–
I/O	46	66	86	K14	229
I/O	47	67	87	L16	232
I/O	–	68	88	M16	235
I/O	–	69	89	L15	238
GND	–	70	90	L14	–
–	–	–	91*	–	–
–	–	–	92*	–	–
–	–	71*	93*	N16*	–
–	–	72*	94*	M15*	–
I/O	48	73	95	P16	241
I/O	49	74	96	M14	244
I/O	–	75	97	N15	247
I/O	–	76	98	P15	250
I/O	50	77	99	N14	253
SGCK3 (I/O)	51	78	100	R16	256
GND	52	79	101	P14	–
–	–	–	102*	–	–
DONE	53	80	103	R15	–
–	–	–	104*	–	–
–	–	–	105*	–	–
VCC	54	81	106	P13	–
–	–	–	107*	–	–
$\overline{PROG}$	55	82	108	R14	–
I/O (D7)	56	83	109	T16	259
PGCK3 (I/O)	57	84	110	T15	262
I/O	–	85	111	R13	265
–	–	–	–	–	–
I/O	–	86	112	P12	268
I/O (D6)	58	87	113	T14	271

Pin Description	PC84	PQ160	PQ208	PG156	Bound Scan
I/O	–	88	114	T13	274
–	–	89*	115*	R12*	–
–	–	89*	115*	R12*	–
–	–	90*	116*	T12*	–
–	–	–	117*	–	–
–	–	–	118*	–	–
GND	–	91	119	P11	–
I/O	–	92	120	R11	277
I/O	–	93	121	T11	280
I/O (D5)	59	94	122	T10	283
I/O (CS0)	60	95	123	P10	286
–	–	–	124*	–	–
–	–	–	125*	–	–
I/O	–	96	126	R10	289
I/O	–	97	127	T9	292
I/O (D4)	61	98	128	R9	295
I/O	62	99	129	P9	298
VCC	63	100	130	R8	–
GND	64	101	131	P8	–
I/O (D3)	65	102	132	T8	301
I/O ($\overline{RS}$)	66	103	133	T7	304
I/O	–	104	134	T6	307
I/O	–	105	135	R7	310
–	–	–	136*	–	–
–	–	–	137*	–	–
I/O (D2)	67	106	138	P7	313
I/O	68	107	139	T5	316
I/O	–	108	140	R6	319
I/O	–	109	141	T4	322
GND	–	110	142	P6	–
–	–	–	143*	–	–
–	–	–	144*	–	–
–	–	111*	145*	R5*	–
–	–	112*	146*	–	–
I/O (D1)	69	113	147	T3	325
I/O (RCLK-$\overline{BUSY}$/RDY)	70	114	148	P5	328
I/O	–	115	149	R4	331
–	–	–	–	–	–
I/O	–	116	150	R3	334
I/O (D0, DIN)	71	117	151	P4	337
SGCK4 (DOUT, I/O)	72	118	152	T2	340
CCLK	73	119	153	R2	–
VCC	74	120	154	P3	–
–	–	–	155*	–	–
–	–	–	156*	–	–
–	–	–	157*	–	–
–	–	–	158*	–	–
O (TDO)	75	121	159	T1	–
GND	76	122	160	N3	–
I/O (A0,$\overline{WS}$)	77	123	161	R1	2
PGCK4 (A1,I/O)	78	124	162	P2	5
I/O	–	125	163	N2	8
–	–	–	–	–	–
I/O	–	126	164	M3	11
I/O (CS1,A2)	79	127	165	P1	14
I/O (A3)	80	128	166	N1	17
–	–	129*	167*	M2*	–
–	–	130*	168*	M1*	–
–	–	–	169*	–	–
–	–	–	170*	–	–
GND	–	131	171	L3	–
I/O	–	132	172	L2	20
I/O	–	133	173	L1	23
I/O (A4)	81	134	174	K3	26
I/O (A5)	82	135	175	K2	29
–	–	–	176*	–	–
–	–	136*	177*	–	–
I/O	–	137	178	K1	32
I/O	–	138	179	J1	35
I/O (A6)	83	139	180	J2	38
I/O (A7)	84	140	181	J3	41
GND	1	141	182	H2	–

* Indicates unconnected package pins.

† Contributes only one bit (.i) to the boundary scan register.

Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 343 = BSCANT.UPD

XC4006 Pinouts

Pin Description	PC84	PG156	PQ160	PQ208	Boundary Scan Order
VCC	2	H3	142	183	-
I/O (A8)	3	H1	143	184	50
I/O (A9)	4	G1	144	185	53
I/O	-	G2	145	186	56
I/O	-	G3	146	187	59
-	-	-	-	188*	-
-	-	-	-	189*	-
I/O (A10)	5	F1	147	190	62
I/O (A11)	6	F2	148	191	65
I/O	-	E1	149	192	68
I/O	-	E2	150	193	72
GND	-	F3	151	194	-
-	-	-	-	195*	-
-	-	-	-	196*	-
I/O	-	D1	152	197	74
I/O	-	D2	153	198	77
I/O (A12)	7	E3	154	199	80
I/O (A13)	8	C1	155	200	83
I/O	-	C2	156	201	86
I/O	-	D3	157	202	89
I/O (A14)	9	B1	158	203	92
SGCK1 (A15, I/O)	10	B2	159	204	95
VCC	11	C3	160	205	-
-	-	-	-	206*	-
-	-	-	-	207*	-
-	-	-	-	208*	-
-	-	-	-	1*	-
GND	12	C4	1	2	-
-	-	-	-	3*	-
PGCK1 (A16, I/O)	13	B3	2	4	98
I/O (A17)	14	A1	3	5	101
I/O	-	A2	4	6	104
I/O	-	C5	5	7	107
I/O (TDI)	15	B4	6	8	110
I/O (TCK)	16	A3	7	9	113
I/O	-	A4	8	10	116
I/O	-	-	9	11	119
-	-	-	-	12*	-
-	-	-	-	13*	-
GND	-	C6	10	14	-
I/O	-	B5	11	15	122
I/O	-	B6	12	16	125
I/O (TMS)	17	A5	13	17	128
I/O	18	C7	14	18	131
-	-	-	-	19*	-
-	-	-	-	20*	-
I/O	-	B7	15	21	136
I/O	-	A6	16	22	137
I/O	19	A7	17	23	140
I/O	20	A8	18	24	143
GND	21	C8	19	25	-
VCC	22	B8	20	26	-

Pin Description	PC84	PG156	PQ160	PQ208	Boundary Scan Order
I/O	23	C9	21	27	146
I/O	24	B9	22	28	149
I/O	-	A9	23	29	152
I/O	-	B10	24	30	155
-	-	-	-	31*	-
-	-	-	-	32*	-
I/O	25	C10	25	33	158
I/O	26	A10	26	34	161
I/O	-	A11	27	35	164
I/O	-	B11	28	36	167
GND	-	C11	29	37	-
-	-	-	-	38*	-
-	-	-	-	39*	-
I/O	-	A12	30	40	170
I/O	-	-	31	41	173
I/O	27	B12	32	42	176
I/O	-	A13	33	43	179
I/O	-	A14	34	44	182
I/O	-	C12	35	45	185
I/O	28	B13	36	46	188
SGCK2 (I/O)	29	B14	37	47	191
M1	30	A15	38	48	194
GND	31	C13	39	49	-
M0	32	A16	40	50	197†
-	-	-	-	51*	-
-	-	-	-	52*	-
-	-	-	-	53*	-
-	-	-	-	54*	-
VCC	33	C14	41	55	-
M2	34	B15	42	56	198†
PGCK2 (I/O)	35	B16	43	57	199
I/O (HDC)	36	D14	44	58	202
I/O	-	C15	45	59	205
I/O	-	D15	46	60	208
I/O	-	E14	47	61	211
I/O ($\overline{\text{LDC}}$)	37	C16	48	62	214
I/O	-	E15	49	63	217
I/O	-	D16	50	64	220
-	-	-	-	65*	-
-	-	-	-	66*	-
GND	-	F14	51	67	-
I/O	-	F15	52	68	223
I/O	-	E16	53	69	226
I/O	38	F16	54	70	229
I/O	39	G14	55	71	232
-	-	-	-	72*	-
-	-	-	-	73*	-
I/O	-	G15	56	74	235
I/O	-	G16	57	75	238
I/O	40	H16	58	76	241
I/O ($\overline{\text{ERR}}$,$\overline{\text{INIT}}$)	41	H15	59	77	244
VCC	42	H14	60	78	-

* Indicates unconnected package pins.

† Contributes only one bit (.i) to the boundary scan register.

XC4006 Pinouts (continued)

Pin Description	PC84	PG156	PQ160	PQ208	Boundary Scan Order
GND	43	J14	61	79	-
I/O	44	J15	62	80	247
I/O	45	J16	63	81	250
I/O	-	K16	64	82	253
I/O	-	K15	65	83	256
-	-	-	-	84*	-
-	-	-	-	85*	-
I/O	46	K14	66	86	259
I/O	47	L16	67	87	262
I/O	-	M16	68	88	265
I/O	-	L15	69	89	268
GND	-	L14	70	90	-
-	-	-	-	91*	-
-	-	-	-	92*	-
I/O	-	N16	71	93	271
I/O	-	M15	72	94	274
I/O	48	P16	73	95	277
I/O	49	M14	74	96	280
I/O	-	N15	75	97	283
I/O	-	P15	76	98	286
I/O	50	N14	77	99	289
SGCK3 (I/O)	51	R16	78	100	292
GND	52	P14	79	101	-
-	-	-	-	102*	-
DONE	53	R15	80	103	-
-	-	-	-	104*	-
-	-	-	-	105*	-
VCC	54	P13	81	106	-
-	-	-	-	107*	-
$\overline{\text{PROG}}$	55	R14	82	108	-
I/O (D7)	56	T16	83	109	295
PGCK3 (I/O)	57	T15	84	110	298
I/O	-	R13	85	111	301
I/O	-	P12	86	112	304
I/O (D6)	58	T14	87	113	307
I/O	-	T13	88	114	310
I/O	-	R12	89	115	313
I/O	-	T12	90	116	316
-	-	-	-	117*	-
-	-	-	-	118*	-
GND	-	P11	91	119	-
I/O	-	R11	92	120	319
I/O	-	T11	93	121	323
I/O (D5)	59	T10	94	122	325
I/O ($\overline{\text{CS0}}$)	60	P10	95	123	328
-	-	-	-	124*	-
-	-	-	-	125*	-
I/O	-	R10	96	126	331
I/O	-	T9	97	127	334
I/O (D4)	61	R9	98	128	337
I/O	62	P9	99	129	340
VCC	63	R8	100	130	-

Pin Description	PC84	PG156	PQ160	PQ208	Boundary Scan Order
GND	64	P8	101	131	-
I/O (D3)	65	T8	102	132	343
I/O ($\overline{\text{RS}}$)	66	T7	103	133	346
I/O	-	T6	104	134	349
I/O	-	R7	105	135	352
-	-	-	-	136*	-
-	-	-	-	137*	-
I/O (D2)	67	P7	106	138	355
I/O	68	T5	107	139	358
I/O	-	R6	108	140	361
I/O	-	T4	109	141	364
GND	-	P6	110	142	-
-	-	-	-	143*	-
-	-	-	-	144*	-
I/O	-	R5	111	145	367
I/O	-	-	112	146	370
I/O (D1)	69	T3	113	147	373
I/O (RCLK-BUSY/$\overline{\text{RDY}}$)	70	P5	114	148	376
I/O	-	R4	115	149	379
I/O	-	R3	116	150	382
I/O (D0, DIN)	71	P4	117	151	385
SGCK4 (DOUT, I/O)	73	T2	118	152	388
CCLK	73	R2	119	153	-
VCC	74	P3	120	154	-
-	-	-	-	155*	-
-	-	-	-	156*	-
-	-	-	-	157*	-
-	-	-	-	158*	-
TDO	75	T1	121	159	-
GND	76	N3	122	160	-
I/O (A0, $\overline{\text{WS}}$)	77	R1	123	161	2
PGCK4 (I/O, A1)	78	P2	124	162	5
I/O	-	N2	125	163	8
I/O	-	M3	126	164	11
I/O (CS1,A2)	79	P1	127	165	14
I/O (A3)	80	N1	128	166	17
I/O	-	M2	129	167	20
I/O	-	M1	130	168	23
-	-	-	-	169*	-
-	-	-	-	170*	-
GND	-	L3	131	171	-
I/O	-	L2	132	172	26
I/O	-	L1	133	173	29
I/O (A4)	81	K3	134	174	32
I/O (A5)	82	K2	135	175	35
-	-	-	-	176*	-
-	-	-	136*	177*	-
I/O	-	K1	137	178	38
I/O	-	J1	138	179	41
I/O (A6)	83	J2	139	180	44
I/O (A7)	84	J3	140	181	47
GND	1	H2	141	182	-

* Indicates unconnected package pins.

Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 391 = BSCAN.UPD

XC4008 Pinouts

Pin Description	PC84	PQ160	PG191	PQ208	Boundary Scan Order
VCC	2	142	J4	183	–
I/O (A8)	3	143	J3	184	56
I/O (A9)	4	144	J2	185	59
I/O	–	145	J1	186	62
I/O	–	146	H1	187	65
I/O	–	–	H2	188	68
I/O	–	–	H3	189	71
I/O (A10)	5	147	G1	190	74
I/O (A11)	6	148	G2	191	77
I/O	–	149	F1	192	80
I/O	–	150	E1	193	83
GND	–	151	G3	194	–
–	–	–	F2*	195*	–
–	–	–	D1*	196*	–
I/O	–	152	C1	197	86
I/O	–	153	E2	198	89
I/O (A12)	7	154	F3	199	92
I/O (A13)	8	155	D2	200	95
I/O	–	156	B1	201	98
–	–	–	–	–	–
I/O	–	157	E3	202	101
I/O (A14)	9	158	C2	203	104
SGCK1 (A15, I/O)	10	159	B2	204	107
VCC	11	160	D3	205	–
–	–	–	–	206*	–
–	–	–	–	207*	–
–	–	–	–	208*	–
–	–	–	–	1*	–
GND	12	1	D4	2	–
–	–	–	–	3*	–
PGCK1 (A16, I/O)	13	2	C3	4	110
I/O (A17)	14	3	C4	5	113
I/O	–	4	B3	6	116
–	–	–	–	–	–
I/O	–	5	C5	7	119
I/O (TDI)	15	6	A2	8	122
I/O (TCK)	16	7	B4	9	125
I/O	–	8	C6	10	128
I/O	–	9	A3	11	131
–	–	–	B5*	12*	–
–	–	–	B6*	13*	–
GND	–	10	C7	14	–
I/O	–	11	A4	15	134
I/O	–	12	A5	16	137
I/O (TMS)	17	13	B7	17	140
I/O	18	14	A6	18	143
I/O	–	–	C8	19	146
I/O	–	–	A7	20	149
I/O	–	15	B8	21	152
I/O	–	16	A8	22	155
I/O	19	17	B9	23	158
I/O	20	18	C9	24	161
GND	21	19	D9	25	–

Pin Description	PC84	PC160	PG191	PQ208	Boundary Scan Order
VCC	22	20	D10	26	–
I/O	23	21	C10	27	164
I/O	24	22	B10	28	167
I/O	–	23	A9	29	170
I/O	–	24	A10	30	173
I/O	–	–	A11	31	176
I/O	–	–	C11	32	179
I/O	25	25	B11	33	182
I/O	26	26	A12	34	185
I/O	–	27	B12	35	188
I/O	–	28	A13	36	191
GND	–	29	C12	37	–
–	–	–	B13*	38*	–
–	–	–	A14*	39*	–
I/O	–	30	A15	40	194
I/O	–	31	C13	41	197
I/O	27	32	B14	42	200
I/O	–	33	A16	43	203
I/O	–	34	B15	44	206
I/O	–	35	C14	45	209
I/O	28	36	A17	46	212
SGCK2 (I/O)	29	37	B16	47	215
M1	30	38	C15	48	218
GND	31	39	D15	49	–
M0	32	40	A18	50	221†
–	–	–	–	51*	–
–	–	–	–	52*	–
–	–	–	–	53*	–
–	–	–	–	54*	–
VCC	33	41	D16	55	–
M2	34	42	C16	56	222†
PGCK2 (I/O)	35	43	B17	57	223
I/O (HDC)	36	44	E16	58	226
–	–	–	–	–	–
I/O	–	45	C17	59	229
I/O	–	46	D17	60	232
I/O	–	47	B18	61	235
I/O (LDC)	37	48	E17	62	238
I/O	–	49	F16	63	241
I/O	–	50	C18	64	244
–	–	–	D18*	65*	–
–	–	–	F17*	66*	–
GND	–	51	G16	67	–
I/O	–	52	E18	68	247
I/O	–	53	F18	69	250
I/O	38	54	G17	70	253
I/O	39	55	G18	71	256
I/O	–	–	H16	72	259
I/O	–	–	H17	73	262
I/O	–	56	H18	74	265
I/O	–	57	J18	75	268
I/O	40	58	J17	76	271
I/O ($\overline{ERR}$, $\overline{INIT}$)	41	59	J16	77	272
VCC	42	60	J15	78	–

* Indicates unconnected package pins.

† Contributes only one bit (.i) to the boundary scan register.

XC4008 Pinouts (continued)

Pin Description	PC84	PQ160	PG191	PQ208	Boundary Scan Order
GND	43	61	K15	79	–
I/O	44	62	K16	80	277
I/O	45	63	K17	81	280
I/O	–	64	K18	82	283
I/O	–	65	L18	83	286
I/O	–	–	L17	84	289
I/O	–	–	L16	85	292
I/O	46	66	M18	86	295
I/O	47	67	M17	87	298
I/O	–	68	N18	88	301
I/O	–	69	P18	89	304
GND	–	70	M16	90	–
–	–	–	N17*	91*	–
–	–	–	R18*	92*	–
I/O	–	71	T18	93	307
I/O	–	72	P17	94	310
I/O	48	73	N16	95	313
I/O	49	74	T17	96	316
I/O	–	75	R17	97	319
I/O	–	76	P16	98	322
I/O	50	77	U18	99	325
SGCK3 (I/O)	51	78	T16	100	328
GND	52	79	R16	101	–
–	–	–	–	102*	–
DONE	53	80	U17	103	–
–	–	–	–	104*	–
–	–	–	–	105*	–
VCC	54	81	R15	106	–
–	–	–	–	107*	–
$\overline{\text{PROG}}$	55	82	V18	108	–
I/O (D7)	56	83	T15	109	331
PGCK3 (I/O)	57	84	U16	110	334
–	–	–	–	–	–
I/O	–	85	T14	111	337
I/O	–	86	U15	112	340
I/O (D6)	58	87	V17	113	343
I/O	–	88	V16	114	346
I/O	–	89	T13	115	349
I/O	–	90	U14	116	352
–	–	–	V15*	117*	–
–	–	–	V14*	118*	–
GND	–	91	T12	119	–
I/O	–	92	U13	120	355
I/O	–	93	V13	121	358
I/O (D5)	59	94	U12	122	361
I/O ($\overline{\text{CSO}}$)	60	95	V12	123	364
I/O	–	–	T11	124	367
I/O	–	–	U11	125	370
I/O	–	96	V11	126	373
I/O	–	97	V10	127	376
I/O (D4)	61	98	U10	128	379
I/O	62	99	T10	129	382
VCC	63	100	R10	130	
GND	64	101	R9	131	

Pin Description	PC84	PQ160	PG191	PQ208	Boundary Scan Order
I/O (D3)	65	102	T9	132	385
I/O ($\overline{\text{RS}}$)	66	103	U9	133	388
I/O	–	104	V9	134	391
I/O	–	105	V8	135	394
I/O	–	–	U8	136	397
I/O	–	–	T8	137	400
I/O (D2)	67	106	V7	138	403
I/O	68	107	U7	139	406
I/O	–	108	V6	140	409
I/O	–	109	U6	141	412
GND	–	110	T7	142	–
–	–	–	V5*	143*	–
–	–	–	V4*	144*	–
I/O	–	111	U5	145	415
I/O	–	112	T6	146	418
I/O (D1)	69	113	V3	147	421
I/O ($\overline{\text{RCLK}}$-$\overline{\text{BUSY}}$/RDY)	70	114	V2	148	426
I/O	–	115	U4	149	427
I/O	–	116	T5	150	430
I/O (D0, DIN)	71	117	U3	151	433
SGCK4 (DOUT, I/O)	72	118	T4	152	436
CCLK	73	119	V1	153	–
VCC	74	120	R4	154	–
–	–	–	–	155*	–
–	–	–	–	156*	–
–	–	–	–	157*	–
–	–	–	–	158*	–
TDO	75	121	U2	159	–
GND	76	122	R3	160	–
I/O (A0, $\overline{\text{WS}}$)	77	123	T3	161	2
PGCK4 (I/O,A1)	78	124	U1	162	5
–	–	–	–	–	–
I/O	–	125	P3	163	8
I/O	–	126	R2	164	11
I/O (CS1, A2)	79	127	T2	165	14
I/O (A3)	80	128	N3	166	17
I/O	–	129	P2	167	20
I/O	–	130	T1	168	23
–	–	–	R1*	169*	–
–	–	–	N2*	170*	–
GND	–	131	M3	171	–
I/O	–	132	P1	172	26
I/O	–	133	N1	173	29
I/O (A4)	81	134	M2	174	32
I/O (A5)	82	135	M1	175	35
I/O	–	–	L3	176	38
I/O	–	136	L2	177	41
I/O	–	137	L1	178	44
I/O	–	138	K1	179	47
I/O (A6)	83	139	K2	180	50
I/O (A7)	84	140	K3	181	53
GND	1	141	K4	182	–

* Indicates unconnected package pins.
Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 439 = BSCAN.UPD

XC4010 Pinouts

Pin Description	PC84	PQ160	PG191	PQ208	CG225	Boundary Scan Order
VCC	2	142	J4	183	A10	-
I/O (A8)	3	143	J3	184	E8	62
I/O (A9)	4	144	J2	185	F8	65
I/O	-	145	J1	186	B7	68
I/O	-	146	H1	187	A7	71
I/O	-	-	H2	188	G7	74
I/O	-	-	H3	189	E7	77
I/O (A10)	5	147	G1	190	F7	80
I/O (A11)	6	148	G2	191	C7	83
I/O	-	149	F1	192	D7	86
I/O	-	150	E1	193	F6	89
GND	-	151	G3	194	A5	-
I/O	-	-	F2	195	B5	92
I/O	-	-	D1	196	D5	96
I/O	-	152	C1	197	C5	98
I/O	-	153	E2	198	C6	101
I/O (A12)	7	154	F3	199	A4	104
I/O (A13)	8	155	D2	200	D4	107
I/O	-	156	B1	201	A3	110
I/O	-	157	E3	202	C2	113
I/O (A14)	9	158	C2	203	D6	116
SGCK1 (A15, I/O)	10	159	B2	204	A2	119
VCC	11	160	D3	205	A6	-
-	-	-	-	206*	-	-
-	-	-	-	207*	-	-
-	-	-	-	208*	-	-
-	-	-	-	1*	-	-
GND	12	1	D4	2	A1	-
-	-	-	-	3*	-	-
PGCK1 (A16, I/O)	13	2	C3	4	B1	122
I/O (A17)	14	3	C4	5	B3	125
I/O	-	4	B3	6	C4	128
I/O	-	5	C5	7	B2	131
I/O (TDI)	15	6	A2	8	C1	134
I/O (TCK)	16	7	B4	9	E3	137
I/O	-	8	C6	10	D2	140
I/O	-	9	A3	11	D3	143
I/O	-	-	B5	12	D1	146
I/O	-	-	B6	13	E5	149
GND	-	10	C7	14	E1	-
I/O	-	11	A4	15	E4	152
I/O	-	12	A5	16	F3	155
I/O (TMS)	17	13	B7	17	F2	158
I/O	18	14	A6	18	F5	161
I/O	-	-	C8	19	G3	164
I/O	-	-	A7	20	G6	167
I/O	-	15	B8	21	G5	170
I/O	-	16	A8	22	G1	173
I/O	19	17	B9	23	H5	176
I/O	20	18	C9	24	H7	179
GND	21	19	D9	25	H1	-
VCC	22	20	D10	26	H2	-
I/O	23	21	C10	27	H6	182
I/O	24	22	B10	28	H3	185
I/O	-	23	A9	29	J6	188

* Indicates unconnected package pins.
† Contributes only one bit (.i) to the boundary scan register.

Pin Description	PC84	PQ160	PG191	PQ208	CG225	Boundary Scan Order
I/O	-	24	A10	30	H4	191
I/O	-	-	A11	31	J1	194
I/O	-	-	C11	32	J5	197
I/O	25	25	B11	33	J3	200
I/O	26	26	A12	34	K2	203
I/O	-	27	B12	35	K5	206
I/O	-	28	A13	36	K3	209
GND	-	29	C12	37	L1	-
I/O	-	-	B13	38	J4	212
I/O	-	-	A14	39	N2	215
I/O	-	30	A15	40	L5	218
I/O	-	31	C13	41	M1	221
I/O	27	32	B14	42	M3	224
I/O	-	33	A16	43	L3	227
I/O	-	34	B15	44	M4	230
I/O	-	35	C14	45	N1	233
I/O	28	36	A17	46	N2	236
SGCK2 (I/O)	29	37	B16	47	K4	239
M1	30	38	C15	48	L4	242
GND	31	39	D15	49	R1	-
M0	32	40	A18	50	P1	245†
-	-	-	-	51*	-	-
-	-	-	-	52*	-	-
-	-	-	-	53*	-	-
-	-	-	-	54*	-	-
VCC	33	41	D16	55	R6	-
M2	34	42	C16	56	R2	246†
PGCK2 (I/O)	35	43	B17	57	P3	247
I/O (HDC)	36	44	E16	58	M6	250
I/O	-	45	C17	59	P2	253
I/O	-	46	D17	60	R3	256
I/O	-	47	B18	61	N3	259
I/O (LDC)	37	48	E17	62	N5	262
I/O	-	49	F16	63	N4	265
I/O	-	50	C18	64	R4	268
I/O	-	-	D18	65	P4	271
I/O	-	-	F17	66	N6	274
GND	-	51	G16	67	R5	-
I/O	-	52	E18	68	M7	277
I/O	-	53	F18	69	P6	280
I/O	38	54	G17	70	L6	283
I/O	39	55	G18	71	N7	286
I/O	-	-	H16	72	P7	289
I/O	-	-	H17	73	M8	291
I/O	-	56	H18	74	R7	295
I/O	-	57	J18	75	N8	298
I/O	40	58	J17	76	J8	301
I/O ($\overline{\text{ERR}}$, $\overline{\text{INIT}}$)	41	59	J16	77	P8	304
VCC	42	60	J15	78	R10	-
GND	43	61	K15	79	R8	-
I/O	44	62	K16	80	L8	307
I/O	45	63	K17	81	M9	310
I/O	-	64	K18	82	P9	313
I/O	-	65	L18	83	R9	316
I/O	-	-	L17	84	K8	319
I/O	-	-	L16	85	L9	322
I/O	46	66	M18	86	P10	325

XC4010 Pinouts (continued)

Pin Description	PC84	PQ160	PG191	PQ208	CG225	Boundary Scan Order
I/O	47	67	M17	87	L10	328
I/O	-	68	N18	88	N10	331
I/O	-	69	P18	89	K10	334
GND	-	70	M16	90	R11	-
I/O	-	-	N17	91	M10	337
I/O	-	-	R18	92	P12	340
I/O	-	71	T18	93	R12	343
I/O	-	72	P17	94	N12	346
I/O	48	73	N16	95	K12	349
I/O	49	74	T17	96	P13	352
I/O	-	75	R17	97	R13	355
I/O	-	76	P16	98	P14	358
I/O	50	77	U18	99	K13	361
SGCK3 (I/O)	51	78	T16	100	M13	364
GND	52	79	R16	101	R15	-
-	-	-	-	102*	-	-
DONE	53	80	U17	103	R14	-
-	-	-	-	104*	-	-
-	-	-	-	105*	-	-
VCC	54	81	R15	106	K15	-
-	-	-	-	107*	-	-
$\overline{PROG}$	55	82	V18	108	P15	-
I/O (D7)	56	83	T15	109	N14	367
PGCK3 (I/O)	57	84	U16	110	L13	370
I/O	-	85	T14	111	N13	373
I/O	-	86	U15	112	N15	376
I/O (D6)	58	87	V17	113	M12	379
I/O	-	88	V16	114	M15	382
I/O	-	89	T13	115	L11	385
I/O	-	90	U14	116	J12	388
I/O	-	-	V15	117	L14	391
I/O	-	-	V14	118	L12	394
GND	-	91	T12	119	L15	-
I/O	-	92	U13	120	K11	397
I/O	-	93	V13	121	H11	400
I/O (D5)	59	94	U12	122	J14	403
I/O ($\overline{CSO}$)	60	95	V12	123	H12	406
I/O	-	-	T11	124	J10	409
I/O	-	-	U11	125	J11	412
I/O	-	96	V11	126	J15	415
I/O	-	97	V10	127	H13	418
I/O (D4)	61	98	U10	128	J9	421
I/O	62	99	T10	129	H9	424
VCC	63	100	R10	130	H14	-
GND	64	101	R9	131	H15	-
I/O ($\overline{D3}$)	65	102	T9	132	H10	427
I/O ($\overline{RS}$)	66	103	U9	133	G12	430
I/O	-	104	V9	134	G14	433
I/O	-	105	V8	135	G15	436
I/O	-	-	U8	136	G9	439
I/O	-	-	T8	137	G11	442
I/O (D2)	67	106	V7	138	G10	445
I/O	68	107	U7	139	G13	448
I/O	-	108	V6	140	F14	451
I/O	-	109	U6	141	F11	454
GND	-	110	T7	142	E15	-
I/O	-	-	V5	143	D14	457

Pin Description	PC84	PQ160	PG191	PQ208	CG225	Boundary Scam Order
I/O	-	-	V4	144	E12	460
I/O	-	111	U5	145	D15	463
I/O	-	112	T6	146	D13	466
I/O (D1)	69	113	V3	147	E13	469
I/O ($\overline{RCLK}$-$\overline{BUSY}$/RDY)	70	114	V2	148	C13	472
I/O	-	115	U4	149	C15	475
I/O	-	116	T5	150	C14	478
I/O (D0, DIN)	71	117	U3	151	D10	481
SGCK4 (I/O)	72	118	T4	152	C11	484
CCLK	73	119	V1	153	D15	-
VCC	74	120	R4	154	F15	-
-	-	-	-	155*	-	-
-	-	-	-	156*	-	-
-	-	-	-	157*	-	-
-	-	-	-	158*	-	-
TD0	75	121	U2	159	A14	-
GND	76	122	R3	160	A15	-
I/O (A0, $\overline{WS}$)	77	123	T3	161	C12	2
PGCK4 (I/O, A1)	78	124	U1	162	C10	5
I/O	-	125	P3	163	B14	8
I/O	-	126	R2	164	A13	11
I/O (CS1, A2)	79	127	T2	165	B13	14
I/O (A3)	80	128	N3	166	B12	17
I/O	-	129	P2	167	E11	20
I/O	-	130	T1	168	D9	23
I/O	-	-	R1	169	B11	26
I/O	-	-	N2	170	D11	29
GND	-	131	M3	171	A11	-
I/O	-	132	P1	172	C9	32
I/O	-	133	N1	173	B10	35
I/O (A4)	81	134	M2	174	B9	38
I/O (A5)	82	135	M1	175	C8	41
I/O	-	-	L3	176	F9	44
I/O	-	136	L2	177	E9	47
I/O	-	137	L1	178	A9	50
I/O	-	138	K1	179	B8	53
I/O (A6)	83	139	K2	180	H8	56
I/O (A7)	84	140	K3	181	G8	59
GND	1	141	K4	182	A8	-

* Indicates unconnected package pins.
Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 487 = BSCAN.UPD

XC4013 Pinouts

Pin Description	MQ208	PG223	CG225	PQ240	Boundary Scan Order
VCC	183	J4	A10	212	-
I/O (A8)	184	J3	E8	213	74
I/O (A9)	185	J2	F8	214	77
I/O	186	J1	B7	215	80
I/O	187	H1	A7	216	83
I/O	188	H2	G7	217	86
I/O	189	H3	E7	218	89
-	-	-	-	219*	-
I/O (A10)	190	G1	F7	220	92
I/O (A11)	191	G2	C7	221	95
VCC	-	-	-	222	-
I/O	-	H4	B6	223	98
I/O	-	G4	E6	224	101
I/O	192	F1	D7	225	104
I/O	193	E1	F6	226	107
GND	194	G3	A5	227	-
I/O	195	F2	B5	228	110
I/O	196	D1	D5	229	113
I/O	197	C1	C5	230	116
I/O	198	E2	C6	231	119
I/O (A12)	199	F3	A4	232	122
I/O (A13)	200	D2	D4	233	125
I/O	-	F4	B4	234	128
I/O	-	E4	C3	235	131
I/O	201	B1	A3	236	134
I/O	202	E3	C2	237	137
I/O (A14)	203	C2	D6	238	140
SGCK1 (A15, I/O)	204	B2	A2	239	143
VCC	205	D3	A6	240	-
-	206*	-	-	-	-
-	207*	-	-	-	-
-	208*	-	-	-	-
-	1*	-	-	-	-
GND	2	D4	A1	1	-
-	3*	-	-	-	-
PGCK1 (A16,I/O)	4	C3	B1	2	146
I/O (A17)	5	C4	B3	3	149
I/O	6	B3	C4	4	152
I/O	7	C5	B2	5	155
I/O (TDI)	8	A2	C1	6	158
I/O (TCK)	9	B4	E3	7	161
I/O	10	C6	D2	8	164
I/O	11	A3	D3	9	167
I/O	12	B5	D1	10	170
I/O	13	B6	E5	11	173
I/O	-	D5	F4	12	176
I/O	-	D6	E2	13	179
GND	14	C7	E1	14	-
I/O	15	A4	E4	15	182
I/O	16	A5	F3	16	185
I/O (TMS)	17	B7	F2	17	188
I/O	18	A6	F5	18	191
VCC	-	-	F1	19	-
I/O	-	D7	G4	20	194
I/O	-	D8	G2	21	197
-	-	-	-	22*	-
I/O	19	C8	G3	23	200
I/O	20	A7	G6	24	203
I/O	21	B8	G5	25	206
I/O	22	A8	G1	26	209
I/O	23	B9	H5	27	212
I/O	24	C9	H7	28	215
GND	25	D9	H1	29	-
VCC	26	D10	H2	30	-

Pin Description	MQ208	PG223	CG225	PQ240	Boundary Scan Order
I/O	27	C10	H6	31	218
I/O	28	B10	H3	32	221
I/O	29	A9	J6	33	224
I/O	30	A10	H4	34	227
I/O	31	A11	J1	35	230
I/O	32	C11	J5	36	233
-	-	-	-	37*	-
I/O	-	D11	J2	38	236
I/O	-	D12	J7	39	239
VCC	-	-	K1	40	-
I/O	33	B11	J3	41	242
I/O	34	A12	K2	42	245
I/O	35	B12	K5	43	248
I/O	36	A13	K3	44	251
GND	37	C12	L1	45	-
I/O	-	D13	K6	46	254
I/O	-	D14	L2	47	257
I/O	38	B13	J4	48	260
I/O	39	A14	M2	49	263
I/O	40	A15	L5	50	266
I/O	41	C13	M1	51	269
I/O	42	B14	H3	52	272
I/O	43	A16	L3	53	275
I/O	44	B15	M4	54	278
I/O	45	C14	N1	55	281
I/O	46	A17	N2	56	284
SGCK2 (I/O)	47	B16	K4	57	287
M1	48	C15	L4	58	290
GND	49	D15	41	59	-
M0	50	A18	P1	60	293†
-	51*	-	-	-	-
-	52*	-	-	-	-
-	53*	-	-	-	-
-	54*	-	-	-	-
VCC	55	D16	R6	61	-
M2	56	C16	R2	62	294†
PGCK2 (I/O)	57	B17	P3	63	295
I/O (HDC)	58	E16	M6	64	298
I/O	59	C17	P2	65	301
I/O	60	D17	R3	66	304
I/O	61	B18	N3	67	307
I/O (LDC)	62	E17	N5	68	310
I/O	63	F16	N4	69	313
I/O	64	C18	R4	70	316
I/O	65	D18	P4	71	319
I/O	66	F17	N6	72	322
I/O	-	E15	P5	73	325
I/O	-	F15	M5	74	328
GND	67	G16	R5	75	-
I/O	68	E18	M7	76	331
I/O	69	F18	P6	77	334
I/O	70	G17	L6	78	337
I/O	71	G18	N7	79	340
VCC	-	-	-	80	-
I/O	72	H16	P7	81	343
I/O	73	H17	M8	82	346
-	-	-	-	83*	-
I/O	-	G15	K7	84	349
I/O	-	H15	L7	85	352
I/O	74	H18	R7	86	355
I/O	75	J18	N8	87	358
I/O	76	J17	J8	88	361
I/O ($\overline{ERR}$, $\overline{INIT}$)	77	J16	P8	89	364
VCC	78	J15	R10	90	-

* Indicates unconnected package pins.
† Contributes only one bit (.i) to the boundary scan register.

XC4013 Pinouts (continued)

Pin Description	MQ208	PG223	CG225	PQ240	Boundary Scan Order
GND	79	K15	R8	91	-
I/O	80	K16	L8	92	367
I/O	81	K17	M9	93	370
I/O	82	K18	P9	94	373
I/O	83	L18	R9	95	376
I/O	84	L17	K8	96	379
I/O	85	L16	L9	97	382
-	-	-	-	98*	-
I/O	-	L15	K9	99	385
I/O	-	M15	N9	100	388
VCC	-	-	-	101	-
I/O	86	M18	P10	102	391
I/O	87	M17	L10	103	394
I/O	88	N18	N10	104	397
I/O	89	P18	K10	105	400
GND	90	M16	R11	106	-
I/O	-	N15	N11	107	403
I/O	-	P15	P11	108	406
I/O	91	N17	M10	109	409
I/O	92	R18	P12	110	412
I/O	93	T18	R12	111	415
I/O	94	P17	N12	112	418
I/O	95	N16	K12	113	421
I/O	96	T17	P13	114	424
I/O	97	R17	R13	115	427
I/O	98	P16	P14	116	430
I/O	99	U18	K13	117	433
SGCK3 (I/O)	100	T16	M13	118	0
GND	101	R16	R15	119	-
-	102*	-	-	-	-
DONE	103	U17	R14	120	-
-	104*	-	-	-	-
-	105*	-	-	-	-
VCC	106	R15	K15	121	-
-	107*	-	-	-	-
$\overline{\text{PROG}}$	108	V18	P15	122	-
I/O (D7)	109	T15	N14	123	439
PGCK3 (I/O)	110	U16	L13	124	442
I/O	111	T14	N13	125	445
I/O	112	U15	N15	126	448
I/O	-	R14	M11	127	451
I/O	-	R13	M14	128	454
I/O (D6)	113	V17	M12	129	457
I/O	114	V16	M15	130	460
I/O	115	T13	L11	131	463
I/O	116	U14	J12	132	466
I/O	117	V15	L14	133	469
I/O	118	V14	L12	134	472
GND	119	T12	L15	135	-
I/O	-	R12	J13	136	475
I/O	-	R11	K14	137	478
I/O	120	U13	K11	138	481
I/O	121	V13	H11	139	484
VCC	-	-	-	140	-
I/O ($\overline{\text{D5}}$)	122	U12	J14	141	487
I/O ($\overline{\text{CSO}}$)	123	V12	H12	142	490
-	-	-	-	143*	-
I/O	124	T11	J10	144	493
I/O	125	U11	J11	145	496
I/O	126	V11	J15	146	499
I/O	127	V10	H13	147	502
I/O (D4)	128	U10	J9	148	505
I/O	129	T10	H9	149	508
VCC	130	R10	H14	150	-

Pin Description	MQ208	PG223	CG225	PQ240	Boundary Scan Order
GND	131	R9	H15	151	-
I/O (D3)	132	T9	H10	152	511
I/O ($\overline{\text{RS}}$)	133	U9	G12	153	514
I/O	134	V9	G14	154	517
I/O	135	V8	G15	155	520
I/O	136	U8	G9	156	523
I/O	137	T8	G11	157	526
-	-	-	-	158*	-
I/O (D2)	138	V7	G10	159	529
I/O	139	U7	G13	160	532
VCC	-	-	-	161	-
I/O	140	V6	F14	162	535
I/O	141	U6	F11	163	538
I/O	-	R8	F13	164	541
I/O	-	R7	F10	165	544
GND	142	T7	E15	166	-
I/O	-	R6	E14	167	547
I/O	-	R5	F12	168	550
I/O	143	V5	D14	169	553
I/O	144	V4	E12	170	556
I/O	145	U5	D15	171	559
I/O	146	T6	D13	172	562
I/O ($\overline{\text{D1}}$)	147	V3	E13	173	565
I/O ($\overline{\text{RCLK}}$-$\overline{\text{BUSY}}$/RDY)	148	V2	C13	174	568
I/O	149	U4	C15	175	571
I/O	150	T5	C14	176	574
I/O (D0, DIN)	151	U3	D10	177	577
SGCK4 (DOUT, I/O)	152	T4	C11	178	580
CCLK	153	V1	B15	179	-
VCC	154	R4	F15	180	-
-	155*	-	-	-	-
-	156*	-	-	-	-
-	157*	-	-	-	-
-	158*	-	-	-	-
TDO	159	U2	A14	181	-
GND	160	R3	A15	182	-
I/O (A0, $\overline{\text{WS}}$)	161	T3	C12	183	2
PGCK4 (I/O, A1)	162	U1	C10	184	5
I/O	163	P3	B14	185	8
I/O	164	R2	A13	186	11
I/O (CS1, A2)	165	T2	B13	187	14
I/O (A3)	166	N3	B12	188	17
I/O	-	P4	D12	189	20
I/O	-	N4	A12	190	23
I/O	167	P2	E11	191	26
I/O	168	T1	D9	192	29
I/O	169	R1	B11	193	32
I/O	170	N2	D11	194	35
-	-	-	-	195*	-
GND	171	M3	A11	196	-
I/O	172	P1	C9	197	38
I/O	173	N1	B10	198	41
I/O	-	M4	E10	199	44
I/O	-	L4	D8	200	47
VCC	-	-	-	201	-
I/O (A4)	174	M2	B9	202	50
I/O (A5)	175	M1	C8	203	53
-	-	-	-	204*	-
I/O	176	L3	F9	205	56
I/O	177	L2	E9	206	59
I/O	178	L1	A9	207	62
I/O	179	K1	B8	208	65
I/O (A6)	180	K2	H8	209	68
I/O (A7)	181	K3	G8	210	71
GND	182	K4	A8	211	-

* Indicates unconnected package pins.

Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 583 = BSCAN.UPD

XC4025 Pinouts

Pin Description	MQ240	PG223	PG299	Bound Scan
VCC	212	J4	K1	-
I/O (A8)	213	J3	K2	98
I/O (A9)	214	J2	K3	101
I/O	215	J1	K5	104
I/O	216	H1	K4	107
I/O	217	H2	J1	110
I/O	218	H3	J2	113
GND	219	-	-	-
I/O (A10)	220	G1	H1	116
I/O (A11)	221	G2	J3	119
I/O	-	-	J4	122
I/O	-	-	J5	125
I/O	-	-	H2	128
I/O	-	-	G1	131
VCC	222	-	E1	-
I/O	223	H4	H3	134
I/O	224	G4	G2	137
I/O	225	F1	H4	140
I/O	226	E1	F2	143
GND	227	G3	F1	-
I/O	-	-	H5	146
I/O	-	-	G3	149
I/O	228	F2	D1	152
I/O	229	D1	G4	155
I/O	230	C1	E2	158
I/O	231	E2	F3	161
I/O (A12)	232	F3	G5	164
I/O (A13)	233	D2	C1	167
I/O	-	-	F4	170
I/O	-	-	E3	173
I/O	234	F4	D2	176
I/O	235	E4	C2	179
I/O	236	B1	F5	182
I/O	237	E3	E4	185
I/O (A14)	238	C2	D3	188
SGCK1 (A15, I/O)	239	B2	C3	191
VCC	240	D3	A2	-
GND	1	D4	B1	-
PGCK1 (A16, I/O)	2	C3	D4	194
VCC	-	-	E5	-
I/O (A17)	3	C4	B2	197
I/O	4	B3	B3	200
I/O	5	C5	E6	203
I/O (TDI)	6	A2	D5	206
I/O (TCK)	7	B4	C4	209
I/O	-	-	A3	212
I/O	-	-	D6	215
I/O	8	C6	E7	218
I/O	9	A3	B4	221
I/O	10	B5	C5	224
I/O	11	B6	A4	227
I/O	12	D5	D7	230
I/O	13	D6	C6	233
I/O	-	-	E8	236
I/O	-	-	B5	239
GND	14	C7	A5	-
I/O	15	A4	B6	242
I/O	16	A5	D8	245
I/O (TMS)	17	B7	C7	248
I/O	18	A6	B7	251
VCC	19	-	A6	-
I/O	20	D7	C8	254
I/O	21	D8	E9	257
GND	22	-	-	-
I/O	-	-	A7	260
I/O	-	-	D9	263
I/O	-	-	B8	266
I/O	-	-	A8	269
I/O	23	C8	C9	272
I/O	24	A7	B9	275
I/O	25	B8	E10	278
I/O	26	A8	A9	281
I/O	27	B9	D10	284
I/O	28	C9	C10	287
GND	29	D9	A10	-
VCC	30	D10	A11	-
I/O	31	C10	B10	290

Pin Description	MQ240	PG223	PG299	Bound Scan
I/O	32	B10	B11	293
I/O	33	A9	C11	296
I/O	34	A10	E11	299
I/O	35	A11	D11	302
I/O	36	C11	A12	305
I/O	-	-	B12	308
I/O	-	-	A13	311
I/O	-	-	C12	314
I/O	-	-	D12	317
GND	37	-	-	-
I/O	38	D11	E12	320
I/O	39	D12	B13	323
VCC	40	-	A16	-
I/O	41	B11	A14	326
I/O	42	A12	C13	329
I/O	43	B12	B14	332
I/O	44	A13	D13	335
GND	45	C12	A15	-
I/O	-	-	B15	338
I/O	-	-	E13	341
I/O	46	D13	C14	344
I/O	47	D14	A17	347
I/O	48	B13	D14	350
I/O	49	A14	B16	353
I/O	50	A15	C15	356
I/O	51	C13	E14	359
I/O	-	-	A18	362
I/O	-	-	D15	365
I/O	52	B14	C16	368
I/O	53	A16	B17	371
I/O	54	B15	B18	374
I/O	55	C14	E15	377
I/O	56	A17	D16	380
SCGK2 (I/O)	57	B16	C17	383
M1	58	C15	A20	386
GND	59	D15	A19	-
M0	60	A18	C18	389†
VCC	61	D16	B20	-
M2	62	C16	D17	390†
PGCK2 (I/O)	63	B17	B19	391
I/O (HDC)	64	E16	C19	394
GND	-	-	E16	-
I/O	65	C17	F16	397
I/O	66	D17	E17	400
I/O	67	B18	D18	403
I/O (LDC)	68	E17	C20	406
I/O	-	-	F17	409
I/O	-	-	G16	412
I/O	69	F16	D19	415
I/O	70	C18	E18	418
I/O	71	D18	D20	421
I/O	72	F17	G17	424
I/O	73	E15	F18	427
I/O	74	F15	H16	430
I/O	-	-	E19	433
I/O	-	-	F19	436
GND	75	G16	E20	-
I/O	76	E18	H17	439
I/O	77	F18	G18	442
I/O	78	G17	G19	445
I/O	79	G18	H18	448
VCC	80	-	F20	-
I/O	81	H16	J16	451
I/O	82	H17	G20	454
GND	83	-	-	-
I/O	-	-	J17	457
I/O	-	-	H19	460
I/O	-	-	H20	463
I/O	-	-	J18	466
I/O	84	G15	J19	469
I/O	85	H15	K16	472
I/O	86	H18	J20	475
I/O	87	J18	K17	478
I/O	88	J17	K18	481
I/O (ERR, INIT)	89	J16	K19	484
VCC	90	J15	L20	-
GND	91	K15	K20	-

† Contributes only one bit (.i) to the boundary scan register.

XC4025 Pinouts (continued)

Pin Description	MQ240	PG223	PG299	Bound Scan
I/O	92	K16	L19	487
I/O	93	K17	L18	490
I/O	94	K18	L16	493
I/O	95	L18	L17	496
I/O	96	L17	M20	499
I/O	97	L16	M19	502
I/O	-	-	N20	505
I/O	-	-	M18	508
I/O	-	-	M17	511
I/O	-	-	M16	514
GND	98	-	-	-
I/O	99	L15	N19	517
I/O	100	M15	P20	520
VCC	101		T20	-
I/O	102	M18	N18	523
I/O	103	M17	P19	526
I/O	104	N18	N17	529
I/O	105	P18	R19	532
GND	106	M16	R20	-
I/O	-	-	N16	535
I/O	-	-	P18	538
I/O	107	N15	U20	541
I/O	108	P15	P17	544
I/O	109	N17	T19	547
I/O	110	R18	R18	550
I/O	111	T18	P16	553
I/O	112	P17	V20	556
I/O	-	-	R17	559
I/O	-	-	T18	562
I/O	113	N16	U19	565
I/O	114	T17	V19	568
I/O	115	R17	R16	571
I/O	116	P16	T17	574
I/O	117	U18	U18	577
SGCK3 (I/O)	118	T16	X20	580
VCC	-	-	T16	-
GND	119	R16	W20	-
DONE	120	U17	V18	-
VCC	121	R16	X19	-
PROG	122	V18	U17	-
I/O (D7)	123	T15	W19	583
PGCK3 (I/O)	124	U16	W18	586
I/O	125	T14	T15	589
I/O	126	U15	U16	592
I/O	127	R14	V17	595
I/O	128	R13	X18	598
I/O	-	-	U15	601
I/O	-	-	T14	604
I/O (D6)	129	V17	W17	607
I/O	130	V16	V16	610
I/O	131	T13	X17	613
I/O	132	U14	U14	616
I/O	133	V15	V15	619
I/O	134	V14	T13	622
I/O	-	-	W16	625
I/O	-	-	W15	628
GND	135	T12	X16	-
I/O	136	R12	U13	631
I/O	137	R11	V14	634
I/O	138	U13	W14	637
I/O	139	V13	V13	640
VCC	140	-	X15	-
I/O (D5)	141	U12	T12	643
I/O (CS0)	142	V12	X14	646
GND	143	-	-	-
I/O	-	-	U12	649
I/O	-	-	W13	652
I/O	-	-	X13	655
I/O	-	-	V12	658
I/O	144	T11	W12	661
I/O	145	U11	T11	664
I/O	146	V11	X12	667
I/O	147	V10	U11	670
I/O (D4)	148	U10	V11	673
I/O	149	T10	W11	676
VCC	150	R10	X10	-
GND	151	R9	X11	-

Pin Description	MQ240	PG223	PG299	Bound Scan
I/O (D3)	152	T9	W10	679
I/O (RS)	153	U9	V10	682
I/O	154	V9	T10	685
I/O	155	V8	U10	688
I/O	156	U8	X9	691
I/O	157	T8	W9	694
I/O	-	-	X8	697
I/O	-	-	V9	700
I/O	-	-	U9	703
I/O	-	-	T9	706
GND	158	-	-	-
I/O (D2)	159	V7	W8	709
I/O	160	U7	X7	712
VCC	161	-	X5	-
I/O	162	V6	V8	715
I/O	163	U6	W7	718
I/O	164	R8	U8	721
I/O	165	R7	W6	724
GND	166	T7	X6	-
I/O	-	-	T8	727
I/O	-	-	V7	730
I/O	167	R6	X4	733
I/O	168	R5	U7	736
I/O	169	V5	W5	739
I/O	170	V4	V6	742
I/O	171	U5	T7	745
I/O	172	T6	X3	748
I/O (D1)	173	V3	U6	751
I/O (RCLK-BUSY/RDY)	174	V2	V5	754
I/O	-	-	W4	757
I/O	-	-	W3	760
I/O	175	U4	T6	763
I/O	176	T5	U5	766
I/O (D0, DIN)	177	U3	V4	769
SGCK4 (DOUT, I/O)	178	T4	X1	772
CCLK	179	V1	V3	-
GND	-	-	T5	-
VCC	180	R4	W1	-
TDO	181	U2	U4	
GND	182	R3	X2	-
I/O (A0, WS)	183	T3	W2	2
PGCK4 (I/O, A1)	184	U1	V2	5
I/O	185	P3	R5	8
I/O	186	R2	T4	11
I/O (CS1, A2)	187	T2	U3	14
I/O (A3)	188	N3	V1	17
I/O	-	-	R4	20
I/O	-	-	P5	23
I/O	189	P4	U2	26
I/O	190	N4	T3	29
I/O	191	P2	U1	32
I/O	192	T1	P4	35
I/O	193	R1	R3	38
I/O	194	N2	N5	41
I/O	195	-	T2	44
I/O	-	-	R2	47
GND	196	M3	T1	-
I/O	197	P1	N4	50
I/O	198	N1	P3	53
I/O	199	M4	P2	56
I/O	200	LV	N3	59
VCC	201	-	R1	-
I/O	-	-	M5	62
I/O	-	-	P1	65
I/O	-	-	M4	68
I/O	-	-	N2	71
I/O (A4)	202	M2	N1	74
I/O (A5)	203	M1	M3	77
GND	204	-	-	-
I/O	205	L3	M2	80
I/O	206	L2	L5	83
I/O	207	L1	M1	86
I/O	208	K1	L4	89
I/O (A6)	209	K2	L3	92
I/O (A7)	210	K3	L2	95
GND	211	K4	L1	-

Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 775 = BSCAN.UPD

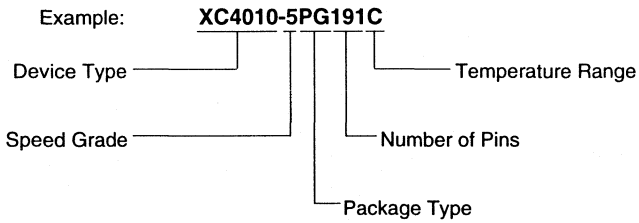
For a detailed description of the device architecture, see pages 2-9 through 2-31.

For a detailed description of the configuration modes and their timing, see pages 2-32 through 2-55.

For detailed lists of package pinouts, see pages 2-57 through 2-67.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS		84	100			120	144	156	160	164	191	196	208		223	225	240		299
TYPE		PLAST. PLCC	PLAST. PQFP	PLAST. VQFP	TOP BRAZED CQFP	CERAM. PGA	PLAST. TQFP	CERAM PGA	PLAST. PQFP	TOP BRAZED CQFP	CERAM. PGA	TOP BRAZED CQFP	PLAST. PQFP	METAL PQFP	CERAM. PGA	CERAM. BGA	PLAST. PQFP	METAL PQFP	METAL PQFP
CODE		PC84	PQ100	VQ100	CB100	PG120	TQ144	PG156	PQ160	CB164	PG191	CB196	PQ208	MQ208	PG223	BG225	PQ240	MQ240	PG299
XC4003	-6	C I	C I			C I													
	-5	C	C			C													
	-4	C	C			C													
XC4005	-10							M B		M B									
	-6	C I						C I M B	C I	M B			C I						
	-5	C I						C I	C I				C I						
	-4	C						C	C				C						
XC4006	-6	C I						C I	C I				C I						
	-5	C I						C I	C I				C I						
	-4	C						C	C				C						
XC4008	-6	C I							C I		C I		(C I)	C I					
	-5	C I							C I		C		(C I)	C I					
	-4	C							C		C		(C)	C					
XC4010	-10										M B	M B							
	-6	C I							C I		C I M B	M B	(C I)	C I		C I			
	-5	C I							C I		C I		(C I)	C I		C I			
	-4	C							C		C		(C)	C		C			
XC4010D	-6	C I							C I										
	-5	C I							C I										
	-4																		
XC4013	-6													C I	C I (M B)	C I	(C I)	C I	
	-5													C	C	C I	(C)	C	
	-4													C	C	C	(C)	C	
XC4025	-6													C I					C I
	-5													C					C
	-4													C					C

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C

B = MIL-STD-883C Class B Parentheses indicate future product plans

XC4010D
Logic Cell Array

Product Specifications

Features

- Third Generation Field-Programmable Gate Array
  - Abundant flip-flops
  - Flexible function generators
  - No on-chip RAM
  - Dedicated high-speed carry-propagation circuit
  - Wide edge decoders (four per edge)
  - Hierarchy of interconnect lines
  - Internal 3-state bus capability
  - Eight global low-skew clock or signal distribution network
- Flexible Array Architecture
  - Programmable logic blocks and I/O blocks
  - Programmable interconnects and wide decoders
- Sub-micron CMOS Process
  - High-speed logic and Interconnect
  - Low power consumption
- Systems-Oriented Features
  - IEEE 1149.1-compatible boundary-scan logic support
  - Programmable output slew rate (2 modes)
  - Programmable input pull-up or pull-down resistors
  - 12-mA sink current per output
  - 24-mA sink current per output pair
- Configured by Loading Binary File
  - Unlimited reprogrammability
  - Six programming modes
- XACT Development System runs on '386/'486-type PC, NEC PC, Apollo, Sun-4, and Hewlett-Packard 700 series
  - Interfaces to popular design environments like Viewlogic, Mentor Graphics and OrCAD
  - Fully automatic partitioning, placement and routing
  - Interactive design editor for design optimization
  - 288 macros, 34 hard macros, RAM/ROM compiler

Description

The XC4010D is a RAM-less, lower-cost version of the XC4010. It is identical to the XC4010 in all respects, except for the missing on-chip RAM.

The XC4010D is available in 84-pin PLCC and in 160-lead Plastic Quad FlatPak packages.

For complete electrical specifications, see pages 2-47 through 2-55.

For a detailed description of the device features, architecture and configuration methods, see pages 2-9 through 2-45.

For detailed lists of package pinouts, see page 2-70.

For package physical dimensions and thermal data, see Section 4.

Ordering Information and Component Availability

XC4010D-6PC84C
XC4010D-6PC84I
XC4010D-5PC84C
XC4010D-5PC84I
XC4010D-6PQ160C
XC4010D-6PQ160I
XC4010D-5PQ160C
XC4010D-5PQ160I

Table 1. The XC4000 Family of Field-Programmable Gate Arrays

Device	XC4003	XC4005	XC4006	XC4008	XC4010/10D	XC4013	XC4025
Appr. Gate Count	3,000	5,000	6,000	8,000	10,000	13,000	25,000
CLB Matrix	10 x 10	14 x 14	16 x 16	18 x 18	20 x 20	24 x 24	32 x 32
Number of CLBs	100	196	256	324	400	576	1,024
Number of Flip-Flops	360	616	768	936	1,120	1,536	2,560
Max Decode Inputs (per side)	30	42	48	54	60	72	96
Max RAM Bits	3,200	6,272	8,192	10,368	12,800*	18,432	32,768
Number of IOBs	80	112	128	144	160	192	256

*XC4010D has no RAM

XC4010D Pinouts

Pin Description	PC84	PQ160	Boundary Scan
VCC	2	142	-
I/O (A8)	3	143	62
I/O (A9)	4	144	65
I/O	-	145	68
I/O	-	146	71
I/O (A10)	5	147	80
I/O (A11)	6	148	83
I/O	-	149	86
I/O	-	150	89
GND	-	151	-
I/O	-	152	98
I/O	-	153	101
I/O (A12)	7	154	104
I/O (A13)	8	155	107
I/O	-	156	110
I/O	-	157	113
I/O (A14)	9	158	116
SGCK1 (A15, I/O)	10	159	119
VCC	11	160	-
GND	12	1	-
PGCK1 (A16, I/O)	13	2	122
I/O (A17)	14	3	125
I/O	-	4	128
I/O	-	5	131
I/O (TDI)	15	6	134
I/O (TCK)	16	7	137
I/O	-	8	140
I/O	-	9	143
GND	-	10	-
I/O	-	11	152
I/O	-	12	155
I/O (TMS)	17	13	158
I/O	18	14	161
I/O	-	15	170
I/O	-	16	173
I/O	19	17	176
I/O	20	18	179
GND	21	19	-
VCC	22	20	-
I/O	23	21	182
I/O	24	22	185
I/O	-	23	188
I/O	-	24	191
I/O	25	25	200
I/O	26	26	203
I/O	-	27	206
I/O	-	28	209
GND	-	29	-
I/O	-	30	218
I/O	-	31	221
I/O	27	32	224
I/O	-	33	227
I/O	-	34	230

Pin Description	PC84	PQ160	Boundary Scan
I/O	-	35	233
I/O	28	36	236
SGCK2 (I/O)	29	37	239
M1	30	38	242
GND	31	39	-
M0	32	40	245†
VCC	33	41	-
M2	34	42	246†
PGCK2 (I/O)	35	43	247
I/O (HDC)	36	44	250
I/O	-	45	253
I/O	-	46	256
I/O	-	47	259
I/O ($\overline{LDC}$)	37	48	262
I/O	-	49	265
I/O	-	50	268
GND	-	51	-
I/O	-	52	277
I/O	-	53	280
I/O	38	54	283
I/O	39	55	286
I/O	-	56	295
I/O	-	57	298
I/O	40	58	301
I/O ($\overline{ERR}$, $\overline{INIT}$)	41	59	304
VCC	42	60	-
GND	43	61	-
I/O	44	62	307
I/O	45	63	310
I/O	-	64	313
I/O	-	65	316
I/O	46	66	325
I/O	47	67	328
I/O	-	68	331
I/O	-	69	334
GND	-	70	-
I/O	-	71	343
I/O	-	72	346
I/O	48	73	349
I/O	49	74	352
I/O	-	75	355
I/O	-	76	358
I/O	50	77	361
SGCK3 (I/O)	51	78	364
GND	52	79	-
DONE	53	80	-
VCC	54	81	-
$\overline{PROG}$	55	82	-
I/O (D7)	56	83	367
PGCK3 (I/O)	57	84	370
I/O	-	85	373
I/O	-	86	376
I/O (D6)	58	87	379

Pin Description	PC84	PQ160	Boundary Scan
I/O	-	88	382
I/O	-	89	385
I/O	-	90	388
GND	-	91	-
I/O	-	92	397
I/O	-	93	400
I/O (D5)	59	94	403
I/O (CS0)	60	95	406
I/O	-	96	415
I/O	-	97	418
I/O (D4)	61	98	421
I/O	62	99	424
VCC	63	100	-
GND	64	101	-
I/O (D3)	65	102	427
I/O (RS)	66	103	430
I/O	-	104	433
I/O	-	105	436
I/O (D2)	67	106	445
I/O	68	107	448
I/O	-	108	451
I/O	-	109	454
GND	-	110	-
I/O	-	111	463
I/O	-	112	466
I/O (D1)	69	113	469
I/O (RCLK-$\overline{BUSY}$/RDY)	70	114	472
I/O	-	115	475
I/O	-	116	478
I/O (D0, DIN)	71	117	481
SGCK4 (DOUT, I/O)	72	118	484
CCLK	73	119	-
VCC	74	120	-
TDO	75	121	-
GND	76	122	-
I/O (A0, $\overline{WS}$)	77	123	2
PGCK4 (I/O, A1)	78	124	5
I/O	-	125	8
I/O	-	126	11
I/O (CS1, A2)	79	127	14
I/O (A3)	80	128	17
I/O	-	129	20
I/O	-	130	23
GND	-	131	-
I/O	-	132	32
I/O	-	133	35
I/O (A4)	81	134	38
I/O (A5)	82	135	41
I/O	-	136	47
I/O	-	137	50
I/O	-	138	53
I/O (A6)	83	139	56
I/O (A7)	84	140	59
GND	1	141	-

Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 487 = BSCAN.UPD

XC4000A
Logic Cell Array Family

Product Specifications

Features

- Third Generation Field-Programmable Gate Arrays
  - Abundant flip-flops
  - Flexible function generators
  - On-chip ultra-fast RAM
  - Dedicated high-speed carry-propagation circuit
  - Wide edge decoders (two per edge)
  - Hierarchy of interconnect lines
  - Internal 3-state bus capability
  - Eight global low-skew clock or signal distribution network
- Flexible Array Architecture
  - Programmable logic blocks and I/O blocks
  - Programmable interconnects and wide decoders
- Sub-micron CMOS Process
  - High-speed logic and Interconnect
  - Low power consumption
- Systems-Oriented Features
  - IEEE 1149.1-compatible boundary-scan logic support
  - Programmable output slew rate (4 modes)
  - Programmable input pull-up or pull-down resistors
  - 24-mA sink current per output (48 per pair)
- Configured by Loading Binary File
  - Unlimited reprogrammability
  - Six programming modes
- XACT Development System runs on '386/'486-type PC, NEC PC, Apollo, Sun-4, and Hewlett-Packard 700 Series
  - Interfaces to popular design environments like Viewlogic, Mentor Graphics and OrCAD
  - Fully automatic partitioning, placement and routing
  - Interactive design editor for design optimization
  - 288 macros, 34 hard macros, RAM/ROM compiler

Description

The XC4000A family of FPGAs offers four devices at the low end of the XC4000 family complexity range. XC4000A differs from XC4000 in four areas: fewer routing resources, fewer wide-edge decoders, higher output sink current, and improved output slew-rate control.

- The XC4000 routing structure is optimized for smaller designs, naturally requiring fewer routing resources. The XC4000A devices have four Longlines and four single-length lines per row and column, while the XC4000 devices have six Longlines and eight single-length lines per row and column. This results in a smaller chip area and lower cost per device.
- XC4000A has two wide-edge decoders on every device edge, while the XC4000 has four. All other wide-decoder features are identical in XC4000 and XC4000A.
- XC4000A outputs are specified at 24 mA, sink current, while XC4000 outputs are specified at 12 mA. The source current is the same 4 mA for both families.
- The XC4000A family offers a more sophisticated output slew-rate control structure with four configurable options for each individual output driver: fast, medium fast, medium slow, and slow. Slew-rate control can alleviate ground-bounce problems when multiple outputs switch simultaneously, and it can reduce or eliminate crosstalk and transmission-line effects on printed circuit boards.

Note that the XC4003 and XC4005 devices are available in both flavors, the lower-priced XC4003A/XC4005A with reduced routing, and the higher-priced XC4003/XC4005 with more abundant routing resources. The XC4000A devices are intended for less demanding and more structured designs, and the XC4000 devices for more random designs requiring additional routing resources.

The equivalent devices are pin-compatible and are available in identical packages, but they are not bitstream compatible. In order to move from a XC4000A to a XC4000, or vice versa, the design must be recompiled.

Table 1. The XC4000A Family of Field-Programmable Gate Arrays

Device	XC4002A	XC4003A	XC4004A	XC4005A
Appr. Gate Count	2,000	3,000	4,000	5,000
CLB Matrix	8 x 8	10 x 10	12 x 12	14 x 14
Number of CLBs	64	100	144	196
Number of Flip-Flops	256	360	480	616
Max Decode Inputs (per side)	24	30	36	42
Max RAM Bits	2,048	3,200	4,608	6,272
Number of IOBs	64	80	96	112

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to 7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to 7	V
V_{TS}	Voltage applied to 3-state output	–0.5 to 7	V
T_{STG}	Storage temperature (ambient)	–65 to + 150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in. = 1.5 mm)	+ 260	°C
T_J	Junction temperature	+ 150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description			Min	Max	Units
V_{CC}	Supply voltage relative to GND	Commercial	0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND	Industrial	–40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND	Military	–55°C to 125°C	4.5	5.5	V
V_{IH}	High-level input voltage (XC4000 has TTL-like input thresholds)			2.0	V_{CC}	V
V_{IL}	Low-level input voltage (XC4000 has TTL-like input thresholds)			0	0.8	V
T_{IN}	Input signal transition time				250	ns

DC Characteristics Over Operating Conditions

Symbol	Description	Min	Max	Units
V_{OH}	High-level output voltage @ I_{OH} = –4.0 mA, V_{CC} min	2.4		V
V_{OL}	Low-level output voltage @ I_{OL} = 24 mA, V_{CC} max (Note 1)		0.4	V
I_{CCO}	Quiescent LCA supply current (Note 2)		10	mA
I_{IL}	Leakage current	–10	+10	µA
C_{IN}	Input capacitance (sample tested)		15	pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0V (sample tested)	0.02	0.25	mA
I_{RLL}	Horizontal Long Line pull-up (when selected) @ logic Low	0.2	2.5	mA

Note: 1. With 50% of the outputs simultaneously sinking 24 mA.
2. With no output current loads, no active input or longline pull-up resistors, all package pins at V_{CC} or GND, and the LCA configured with a MakeBits tie option.

Wide Decoder Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, there derived from benchmark timing patterns. The following guidelines relflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
Full length, both pull-ups, inputs from IOB I-pins	T_{WAF}	XC4002A	8.5	7.5	4.5	ns
		XC4003A	9.0	8.0	5.0	ns
		XC4004A	9.5	8.5	5.5	ns
		XC4005A	10.0	9.0	6.0	ns
Full length, both pull-ups inputs from internal logic	T_{WAFL}	XC4002A	11.5	10.5	6.5	ns
		XC4003A	12.0	11.0	7.0	ns
		XC4004A	12.5	11.5	7.5	ns
		XC4005A	13.0	12.0	8.0	ns
Half length, one pull-up inputs from IOB I-pins	T_{WAO}	XC4002A	8.5	7.5	5.5	ns
		XC4003A	9.0	8.0	6.0	ns
		XC4004A	9.5	8.5	6.5	ns
		XC4005A	10.0	9.0	7.0	ns
Half length, one pull-up inputs from internal logic	T_{WAOL}	XC4002A	11.5	10.5	7.5	ns
		XC4003A	12.0	11.0	8.0	ns
		XC4004A	12.5	11.5	8.5	ns
		XC4005A	13.0	12.0	9.0	ns

Note: These delays are specified from the decoder input to the decoder output. For pin-to-pin delays, add the input delay (T_{PID}) and output delay (one of 4 modes), as listed on page 2-70.

Global Buffer Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
Global Signal Distribution						
From pad through **primary** buffer, to any clock k	T_{PG}	XC4002A	7.7	5.7	4.9	ns
		XC4003A	7.8	5.8	5.1	ns
		XC4004A	7.9	5.9	5.3	ns
		XC4005A	8.0	6.0	5.5	ns
From pad through **secondary** buffer, to any clock k	T_{SG}	XC4002A	8.7	6.7	6.1	ns
		XC4003A	8.8	6.8	6.3	ns
		XC4004A	8.9	6.9	6.5	ns
		XC4005A	9.0	7.0	6.7	ns

Horizontal Longline Switching Characteristic Guidelines

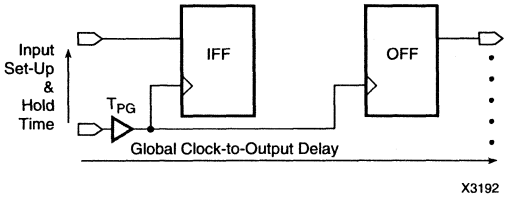
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
TBUF driving a Horizontal Longline (L.L.) I going High or Low to L.L. going High or Low, while T is Low, i.e. buffer is constantly active	T_{IO1}	XC4002A	8.2	6.0	4.0	ns
		XC4003A	8.8	6.2	4.4	ns
		XC4004A	9.4	6.6	5.0	ns
		XC4005A	10.0	7.0	5.5	ns
I going Low to L.L. going from resistive pull-up High to active Low, (TBUF configured as open drain)	T_{IO2}	XC4002A	8.7	6.5	4.5	ns
		XC4003A	9.3	6.7	5.0	ns
		XC4004A	9.9	7.1	5.5	ns
		XC4005A	10.5	7.5	6.0	ns
T going Low to L.L. going from resistive pull-up or floating High to active Low, (TUBF configured as open drain)	T_{ON}	XC4002A	10.1	8.4	6.8	ns
		XC4003A	10.7	9.0	7.2	ns
		XC4004A	11.4	9.5	7.6	ns
		XC4005A	12.0	10.0	8.0	ns
T going High to TBUF going inactive, not driving L.L.	T_{OFF}	All devices	3.0	2.0	1.8	ns
T going High to L.L. going from Low to High, pulled up by a single resistor	T_{PUS}	XC4002A	23.0	19.0	13.0	ns
		XC4003A	24.0	20.0	14.0	ns
		XC4004A	25.0	21.0	15.0	ns
		XC4005A	26.0	22.0	16.0	ns
T going High to L.L. going from Low to High, pulled up by two resistors	T_{PUF}	XC4002A	10.5	8.5	6.5	ns
		XC4003A	11.0	9.0	7.0	ns
		XC4004A	11.5	9.5	7.5	ns
		XC4005A	12.0	10.0	8.0	ns

Guaranteed Input and Output Parameters (Pin-to-Pin)

All values listed below are tested directly. and guaranteed over the operating conditions. The same parameters can also be derived indirectly from the IOB and Global Buffer specifications. The XACT delay calculator uses this indirect method. When there is a discrepancy between these two methods, the directly tested values listed below should be used, and the derived values should be ignored.

		Speed Grade	-6	-5	-4	
Description	**Symbol**	**Device**				**Units**
Global Clock to Output (fast)	T_{ICKOF}	XC4002A	14.9	12.2	11.4	ns
		XC4003A	15.1	12.5	11.6	ns
	(Max)	XC4004A	15.3	12.8	11.8	ns
		XC4005A	15.5	13.0	12.0	ns
Global Clock to Output (slew limited)	T_{ICKO}	XC4002A	19.9	15.2	14.4	ns
		XC4003A	20.1	15.5	14.6	ns
	(Max)	XC4004A	20.3	15.8	14.8	ns
		XC4005A	20.5	16.0	15.0	ns
Input Set-up Time, using IFF (fast)	T_{PSUF}	XC4002A	2.6	2.3	1.8	ns
		XC4003A	2.4	2.0	1.6	ns
	(Min)	XC4004A	2.2	1.7	1.4	ns
		XC4005A	2.0	1.5	1.2	ns
Input Hold time, using IFF (fast)	T_{PHF}	XC4002A	4.9	3.7	3.7	ns
		XC4003A	5.1	4.0	4.0	ns
	(Min)	XC4004A	5.3	4.3	4.3	ns
		XC4005A	5.5	4.5	4.5	ns
Input Set-up Time, using IFF (with delay)	T_{PSU}	XC4002A	21.8	18.8	12.0	ns
		XC4003A	21.5	18.5	12.0	ns
	(Min)	XC4004A	21.2	18.2	12.0	ns
		XC4005A	21.0	18.0	12.0	ns
Input Hold Time, using IFF (with delay)	T_{PH}	XC4002A	0	0	0	ns
		XC4003A	0	0	0	ns
	(Min)	XC4004A	0	0	0	ns
		XC4005A	0	0	0	ns



Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture). When testing fast outputs, only one output switches. When testing slew-rate limited outputs, half the number of outputs on one side of the device are switching. These parameter values are tested and guaranteed for worst-case conditions of supply voltage and temperature, and also with the most unfavorable clock polarity choice.

T_{PDLI} for -4 Speed Grade

Pad to I1, I2	XC4002A	17.4 ns
via transparent	XC4003A	17.6 ns
latch, with delay	XC4004A	17.8 ns
PRELIMINARY	XC4005A	17.9 ns

T_{PICKD} for -4 Speed Grade

Input set-up time	XC4002A	15.4 ns
pad to clock (IK)	XC4003A	15.6 ns
with delay	XC4004A	15.8 ns
PRELIMINARY	XC4005A	15.9 ns

X5283

IOB Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

		-6		-5		-4		
Description	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
INPUT								
Propagation Delays								
Pad to I1, I2	T_{PID}		4.0		3.0		2.8	ns
Pad to I1, I2, via transparent latch (fast)	T_{PLI}		8.0		7.0		6.0	ns
Pad to I1, I2, via transparent latch (with delay)	T_{PDLI}		26.0		24.0		**	ns
Clock (IK) toI1, I2, (flip-flop)	T_{IKRI}		8.0		7.0		6.0	ns
Clock (IK) to I1, I2 (latch enable, active Low)	T_{IKLI}		8.0		7.0		6.0	ns
Set-up Time (Note 3)								
Pad to Clock (IK), fast	T_{PICK}	7.0		6.0		4.0		ns
Pad to Clock (IK) with delay	T_{PICKD}	25.0		24 .0		**		ns
Hold Time (Note 3)								
Pad to Clock (IK), fast	T_{IKPI}	1.0		1.0		1.0		ns
Pad to Clock (IK) with delay	T_{IKPID}	neg		neg		neg		ns
OUTPUT								
Propagation Delays								
Clock (OK) to Pad (fast)	T_{OKPOF}		7.5		7.0		6.5	ns
Output (O) to Pad (fast)	T_{OPF}		9.0		7.0		5.5	ns
3-state to Pad begin hi-Z (slew-rate independent)	T_{TSHZ}		9.0		7.0		6.5	ns
3-state to Pad active and valid (fast)	T_{TSONF}		13.0		10.0		9.5	ns
Additional Delay								
For medium fast outputs			2.0		1.5		1.0	ns
For medium slow outputs			4.0		3.0		2.0	ns
For slow outputs			6.0		4.5		3.0	ns
Set-up and Hold Times								
Output (O) to clock (OK) set-up time	T_{OOK}	8.0		6.0		5.5		ns
Output (O) to clock (OK) hold time	T_{OKO}	0.0		0.0		0		ns
Clock								
Clock High or Low time	T_{CH}/T_{CL}	5.0		4.0		4.0		ns
Global Set/Reset								
Delay from GSR net through Q to I1, I2	T_{RRI}		14.5		13.5		13.5	ns
Delay from GSR net to Pad	T_{RPO}		18.0		17.0		14.6	ns
GSR width*	T_{MRW}	21.0		18.0		18.0		ns

* Timing is based on the XC4005. For other devices see XACT timing calculator.
** See preceding page.

Notes: 1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture).

2. Voltage levels of unused (bonded and unbonded) pads must be valid logic levels. Each can be configured with the internal pull-up or pull-down resistor or alternatively configured as a driven output or be driven from an external source.

3. Input pad setup times and hold times are specified with respect to the internal clock (IK). To calculate system setup time, subtract clock delay (clock pad to IK) from the specified input pad setup time value, but do not subtract below zero. Negative hold time means that the delay in the input data is adequate for the **external system hold time** to be zero, provided the input clock uses the Global signal distribution from pad to IK.

CLB Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

Speed Grade		-6		-5		-4		
Description	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Combinatorial Delays								
F/G inputs to X/Y outputs	T_{ILO}		6.0		4.5		4.0	ns
F/G inputs via H' to X/Y outputs	T_{IHO}		8.0		7.0		6.0	ns
C inputs via H' to X/Y outputs	T_{HHO}		7.0		5.0		4.5	ns
CLB Fast Carry Logic								
Operand inputs (F1,F2,G1,G4) to C_{OUT}	T_{OPCY}		7.0		5.5		5.0	ns
Add/Subtract input (F3) to C_{OUT}	T_{ASCY}		8.0		6.0		5.5	ns
Initialization inputs (F1,F3) to C_{OUT}	T_{INCY}		6.0		4.0		3.5	ns
C_{IN} through function generators to X/Y outputs	T_{SUM}		8.0		6.0		5.5	ns
C_{IN} to C_{OUT}, bypass function generators.	T_{BYP}		2.0		1.5		1.5	ns
Sequential Delays								
Clock K to outputs Q	T_{CKO}		5.0		3.0		3.0	ns
Set-up Time before Clock K								
F/G inputs	T_{ICK}	6.0		4.5		4.5		ns
F/G inputs via H'	T_{IHCK}	8.0		6.0		6.0		ns
C inputs via H1	T_{HHCK}	7.0		5.0		5.0		ns
C inputs via DIN	T_{DICK}	4.0		3.0		3.0		ns
C inputs via EC	T_{ECCK}	7.0		4.0		3.0		ns
C inputs via S/R, going Low (inactive)	T_{RCK}	6.0		4.5		4.0		ns
C_{IN} input via F'/G'		8.0		6.0		5.5		ns
C_{IN} input via F'/G' and H'		10.0		7.5		7.3		ns
Hold Time after Clock K								
F/G inputs	T_{CKI}	0		0		0		ns
F/G inputs via H'	T_{CKIH}	0		0		0		ns
C inputs via H1	T_{CKHH}	0		0		0		ns
C inputs via DIN	T_{CKDI}	0		0		0		ns
C inputs via EC	T_{CKEC}	0		0		0		ns
C inputs via S/R, going Low (inactive)	T_{CKR}	0		0		0		ns
Clock								
Clock High time	T_{CH}	5.0		4.0		4.0		ns
Clock Low time	T_{CL}	5.0		4.0		4.0		ns
Set/Reset Direct								
Width (High)	T_{RPW}	5.0		4.0		4.0		ns
Delay from C inputs via S/R, going High to Q	T_{RIO}		9.0		8.0		7.0	ns
Master Set/Reset*								
Width (High or Low)	T_{MRW}	21.0		18.0		18.0		ns
Delay from Global Set/Reset net to Q	T_{MRQ}		33.0		31.0		28.0	ns

* Timing is based on the XC4005. For other devices see XACT timing calculator.

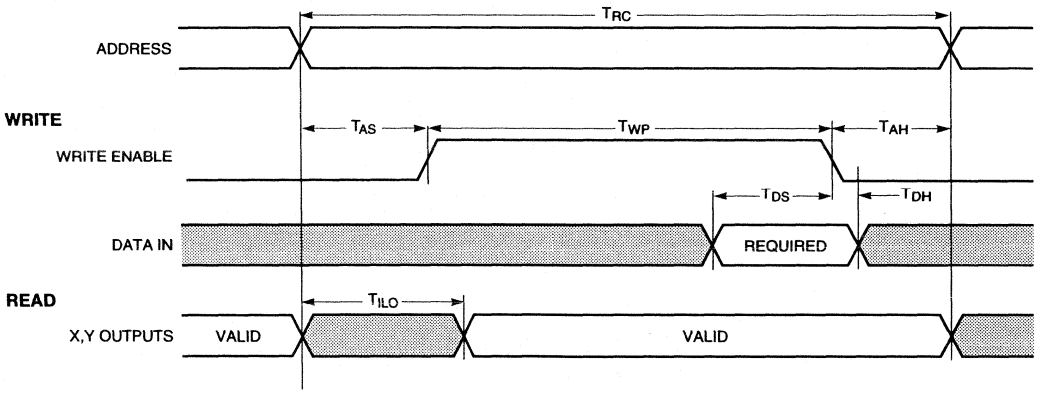
CLB Switching Characteristic Guidelines (continued)

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

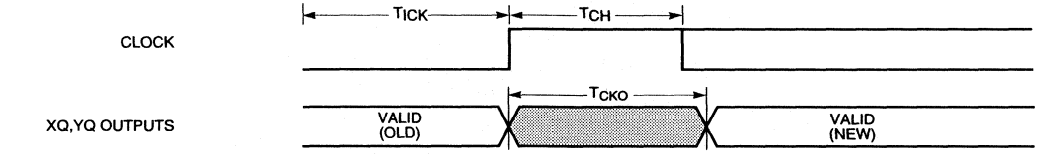
CLB RAM OPTION	**Speed Grade**		**-6**		**-5**		**-4**		
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Write Operation									
Address write cycle time	16 x 2	T_{WC}	9.0		8.0		8.0		ns
	32 x 1	T_{WCT}	9.0		8.0		8.0		ns
Write Enable pulse width (High)	16 x 2	T_{WP}	5.0		4.0		4.0		ns
	32 x 1	T_{WPT}	5.0		4.0		4.0		ns
Address set-up time before beginning of WE	16 x 2	T_{AS}	2.0		2.0		2.0		ns
	32 x 1	T_{AST}	2.0		2.0		2.0		ns
Address hold time after end of WE	16 x 2	T_{AH}	2.0		2.0		2.0		ns
	32 x 1	T_{AHT}	2.0		2.0		2.0		ns
DIN set-up time before end of WE	16 x 2	T_{DS}	4.0		4.0		4.0		ns
	32 x 1	T_{DST}	5.0		5.0		5.0		ns
DIN hold time after end of WE	both	T_{DHT}	2.0		2.0		2.0		ns
Read Operation									
Address read cycle time	16 x 2	T_{RC}	7.0		5.5		5.0		ns
	32 x 1	T_{RCT}	10.0		7.5		7.0		ns
Data valid after address change	16 x 2	T_{ILO}		6.0		4.5		4.0	ns
(no Write Enable)	32 x 1	T_{IHO}		8.0		7.0		6.0	ns
Read Operation, Clocking Data into Flip-Flop									
Address setup time before clock K	16 x 2	T_{ICK}	6.0		4.5		4.5		ns
	32 x 1	T_{IHCK}	8.0		6.0		6.0		ns
Read During Write									
Data valid after WE going active	16 x 2	T_{WO}		12.0		10.0		9.0	ns
(DIN stable before WE)	32 x 1	T_{WOT}		15.0		12.0		11.0	ns
Data valid after DIN	16 x 2	T_{DO}		11.0		9.0		8.5	ns
(DIN change during WE)	32 x 1	T_{DOT}		14.0		11.0		11.0	ns
Read During Write, Clocking Data into Flip-Flop									
WE setup time before clock K	16 x 2	T_{WCK}	12.0		10.0		9.5		ns
	32 x 1	T_{WCKT}	15.0		12.0		11.5		ns
Data setup time before clock K	16 x 2	T_{DCK}	11.0		9.0		9.0		ns
	32 x 1	T_{DCKT}	14.0		11.0		11.0		ns

Note: Timing for the 16 x 1 RAM option is identical to 16 x 2 RAM timing

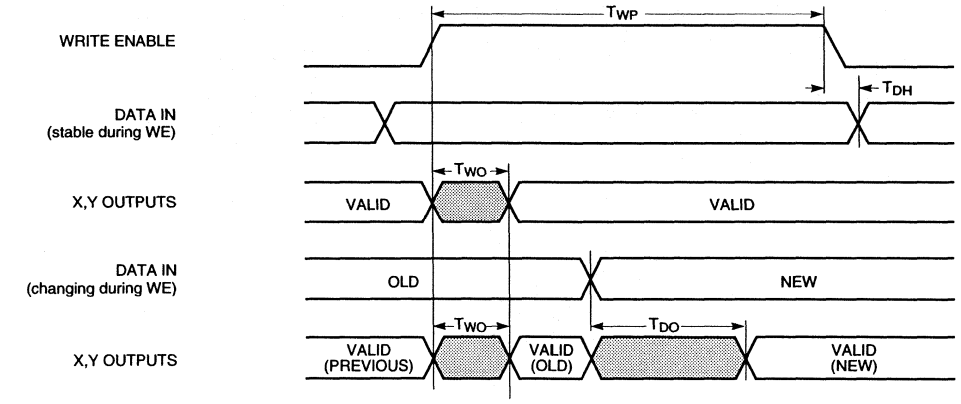
CLB RAM Timing Characteristics



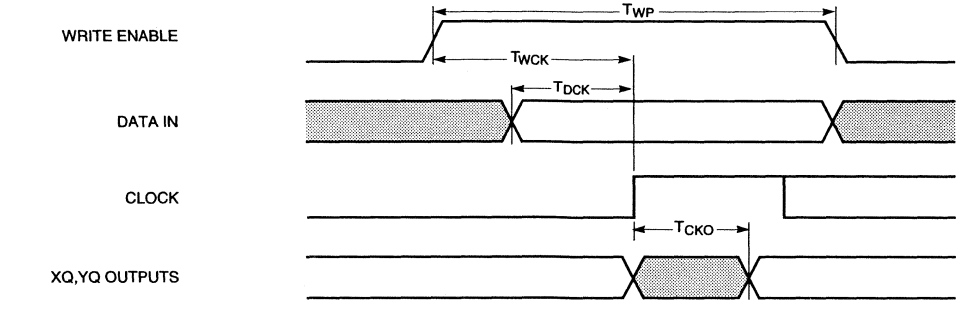
READ, CLOCKING DATA INTO FLIP-FLOP



READ DURING WRITE



READ DURING WRITE, CLOCKING DATA INTO FLIP-FLOP



X2640


XILINX
XC4013™ -5
PG223C
X08701E
X9243MG
XILINX
XC4005-5
PC156C
X05671E
X9139MG
XILINX®
XC4013™ -5
MQ240C
X30339M
BK19353
XILINX
XC4003H™ -5
PG191C
X08700E
X9243MJ
XILINX®
XC4010™ -5
CG225C
X30010M
BK19401

XC4002A Pinouts

Pin Description	PC 84	PQ100	VQ100	PG120	Bound Scan
VCC	2	92	89	G3	–
I/O (A8)	3	93	90	G1	26
I/O (A9)	4	94	91	F1	29
–	–	95*	92*	E1*	–
–	–	96*	93*	F2*	–
I/O (A10)	5	97	94	F3	32
I/O (A11)	6	98	95	D1	35
–	–	–	–	E2*	–
I/O (A12)	7	99	96	C1	38
I/O (A13)	8	100	97	D2	41
–	–	–	–	E3*	–
–	–	–	–	B1*	–
I/O (A14)	9	1	98	C2	44
SGCK1 (A15, I/O)	10	2	99	D3	47
VCC	11	3	100	C3	–
GND	12	4	1	C4	–
PGCK1 (A16, I/O)	13	5	2	B2	50
I/O (A17)	14	6	3	B3	53
–	–	–	–	A1*	–
–	–	–	–	A2*	–
I/O (TDI)	15	7	4	C5	56
I/O (TCK)	16	8	5	B4	59
–	–	–	–	A3*	–
I/O (TMS)	17	9	6	B5	62
I/O	18	10	7	A4	65
–	–	–	–	C6*	–
–	–	11*	8*	A5*	–
I/O	19	12	9	B6	68
I/O	20	13	10	A6	71
GND	21	14	11	B7	–
VCC	22	15	12	C7	–
I/O	23	16	13	A7	74
I/O	24	17	14	A8	77
–	–	18*	15*	A9*	–
–	–	–	–	B8*	–
I/O	25	19	16	C8	80
I/O	26	20	17	A10	83
I/O	27	21	18	B9	86
I/O	–	22	19	A11	89
–	–	–	–	B10*	–

Pin Description	PC 84	PQ100	VQ100	PG120	Bound Scan
I/O	28	23	20	C9	92
SGCK2 (I/O)	29	24	21	A12	95
O (M1)	30	25	22	B11	98
GND	31	26	23	C10	–
I (M0)	32	27	24	C11	101†
VCC	33	28	25	D11	–
I (M2)	34	29	26	B12	102†
PGCK2 (I/O)	35	30	27	C12	103
I/O (HDC)	36	31	28	A13	106
–	–	–	–	B13*	–
–	–	–	–	E11*	–
I/O	–	32	29	D12	109
I/O ($\overline{LDC}$)	37	33	30	C13	112
I/O	38	34	31	E12	115
I/O	39	35	32	D13	118
–	–	36*	33*	F11*	–
–	–	37*	34*	E13*	–
I/O	40	38	35	F12	121
I/O ($\overline{ERR}$, $\overline{INIT}$)	41	39	36	F13	124
VCC	42	40	37	G12	–
GND	43	41	38	G11	–
I/O	44	42	39	G13	127
I/O	45	43	40	H13	130
–	–	44*	41*	J13*	–
–	–	45*	42*	H12*	–
I/O	46	46	43	H11	133
I/O	47	47	44	K13	136
I/O	48	48	45	J12	139
I/O	49	49	46	L13	142
–	–	–	–	K12*	–
–	–	–	–	J11*	–
I/O	50	50	47	M13	145
SGCK3 (I/O)	51	51	48	L12	148
GND	52	52	49	K11	–
DONE	53	53	50	L11	–
VCC	54	54	51	L10	–
$\overline{PROG}$	55	55	52	M12	–
I/O (D7)	56	56	53	M11	151
PGCK3 (I/O)	57	57	54	N13	154
–	–	–	–	N12*	–

Pin Description	PC 84	PQ100	VQ100	PG120	Bound Scan
–	–	–	–	L9	–
I/O (D6)	58	58	55	M10	157
I/O	–	59	56	N11	160
I/O (D5)	59	60	57	M9	163
I/O ($\overline{CSO}$)	60	61	58	N10	166
–	–	62*	59*	L8*	–
–	–	63*	60*	N9*	–
I/O (D4)	61	64	61	M8	169
I/O	62	65	62	N8	172
VCC	63	66	63	M7	–
GND	64	67	64	L7	–
I/O (D3)	65	68	65	N7	175
I/O ($\overline{RS}$)	66	69	66	N6	178
–	–	70*	67*	N5*	–
–	–	–	–	M6*	–
I/O (D2)	67	71	68	L6	181
I/O	68	72	69	N4	184
I/O (D1)	69	73	70	M5	187
I/O (RCLK-$\overline{BUSY}$/RDY)	70	74	71	N3	190
–	–	–	–	M4*	–
–	–	–	–	L5*	–
I/O (D0, DIN)	71	75	72	N2	193
SGCK4 (DOUT, I/O)	72	76	73	M3	196
CCLK	73	77	74	L4	–
VCC	74	78	75	L3	–
O (TDO)	75	79	76	M2	–
GND	76	80	77	K3	–
I/O (A0, $\overline{WS}$)	77	81	78	L2	2
PGCK4 (I/O,A1)	78	82	79	N1	5
–	–	–	–	M1*	–
–	–	–	–	J3*	–
I/O (CS1, A2)	79	83	80	K2	8
I/O (A3)	80	84	81	L1	11
I/O (A4)	81	85	82	J2	14
I/O (A5)	82	86	83	K1	17
–	–	87*	84*	H3*	–
–	–	88*	85*	J1*	–
I/O (A6)	83	89	86	H2	20
I/O (A7)	84	90	87	H1	23
GND	1	91	88	G2	–

* Indicates unconnected package pins.
† Contributes only one bit (.i) to the boundary scan register.
Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 199 = BSCANT.UPD

XC4003A Pinouts

Pin Description	PC84	VQ100	PQ100	PG120	Bound Scan
VCC	2	89	92	G3	–
I/O (A8)	3	90	93	G1	32
I/O (A9)	4	91	94	F1	35
I/O	–	92	95	E1	38
I/O	–	93	96	F2	41
I/O (A10)	5	94	97	F3	44
I/O (A11)	6	95	98	D1	47
–	–	–	–	E2*	–
I/O (A12)	7	96	99	C1	50
I/O (A13)	8	97	100	D2	53
–	–	–	–	E3*	–
–	–	–	–	B1*	–
I/O (A14)	9	98	1	C2	56
SGCK1 (A15,I/O)	10	99	2	D3	59
VCC	11	100	3	C3	–
GND	12	1	4	C4	–
PGCK1 (A16, I/O)	13	2	5	B2	62
I/O (A17)	14	3	6	B3	65
–	–	–	–	A1*	–
–	–	–	–	A2*	–
I/O (TDI)	15	4	7	C5	68
I/O (TCK)	16	5	8	B4	71
–	–	–	–	A3*	–
I/O (TMS)	17	6	9	B5	74
I/O	18	7	10	A4	77
I/O	–	–	–	C6	80
I/O	–	8	11	A5	83
I/O	19	9	12	B6	86
I/O	20	10	13	A6	89
GND	21	11	14	B7	–
VCC	22	12	15	C7	–
I/O	23	13	16	A7	92
I/O	24	14	17	A8	95
I/O	–	15	18	A9	98
I/O	–	–	–	B8	101
I/O	25	16	19	C8	104
I/O	26	17	20	A10	107
I/O	27	18	21	B9	110
I/O	–	19	22	A11	113
–	–	–	–	B10*	–
I/O	28	20	23	C9	116
SGCK2 (I/O)	29	21	24	A12	119
O (M1)	30	22	25	B11	122
GND	31	23	26	C10	–
I (M0)	32	24	27	C11	125†
VCC	33	25	28	D11	–
I (M2)	34	26	29	B12	126†
PGCK2 (I/O)	35	27	30	C12	127
I/O (HDC)	36	28	31	A13	130
–	–	–	–	B13*	–
–	–	–	–	E11*	–
I/O	–	29	32	D12	133
I/O ($\overline{\text{LDC}}$)	37	30	33	C13	136
I/O	38	31	34	E12	139
I/O	39	32	35	D13	142
I/O	–	33	36	F11	145
I/O	–	34	37	E13	148
I/O	40	35	38	F12	151
I/O ($\overline{\text{ERR}}$, $\overline{\text{INIT}}$)	41	36	39	F13	154
VCC	42	37	40	G12	–

Pin Description	PC84	VQ100	PQ100	PG120	Bound Scan
GND	43	38	41	G11	–
I/O	44	39	42	G13	157
I/O	45	40	43	H13	160
I/O	–	41	44	J13	163
I/O	–	42	45	H12	166
I/O	46	43	46	H11	169
I/O	47	44	47	K13	172
I/O	48	45	48	J12	175
I/O	49	46	49	L13	178
–	–	–	–	K12*	–
–	–	–	–	J11*	–
I/O	50	47	50	M13	181
SGCK3 (I/O)	51	48	51	L12	184
GND	52	49	52	K11	–
DONE	53	50	53	L11	–
VCC	54	51	54	L10	–
$\overline{\text{PROG}}$	55	52	55	M12	–
I/O (D7)	56	53	56	M11	187
PGCK3 (I/O)	57	54	57	N13	190
–	–	–	–	N12*	–
–	–	–	–	L9*	–
I/O (D6)	58	55	58	M10	193
I/O	–	56	59	N11	196
I/O (D5)	59	57	60	M9	199
I/O ($\overline{\text{CS0}}$)	60	58	61	N10	202
I/O	–	59	62	L8	205
I/O	–	60	63	N9	208
I/O (D4)	61	61	64	M8	211
I/O	62	62	65	N8	214
VCC	63	63	66	M7	–
GND	64	64	67	L7	–
I/O (D3)	65	65	68	N7	217
I/O ($\overline{\text{RS}}$)	66	66	69	N6	220
I/O	–	67	70	N5	223
I/O	–	–	–	M6	226
I/O (D2)	67	68	71	L6	229
I/O	68	69	72	N4	232
I/O (D1)	69	70	73	M5	235
I/O (RCLK-$\overline{\text{BUSY}}$/RDY)	70	71	74	N3	238
–	–	–	–	M4*	–
–	–	–	–	L5*	–
I/O (D0, DIN)	71	72	75	N2	241
SGCK4 (DOUT, I/O)	72	73	76	M3	244
CCLK	73	74	77	L4	–
VCC	74	75	78	L3	–
O (TDO)	75	76	79	M2	–
GND	76	77	80	K3	–
I/O (A0, $\overline{\text{WS}}$)	77	78	81	L2	2
PGCK4 (A1, I/O)	78	79	82	N1	5
–	–	–	–	M1*	–
–	–	–	–	J3*	–
I/O (CS1, A2)	79	80	83	K2	8
I/O (A3)	80	81	84	L1	11
I/O (A4)	81	82	85	J2	14
I/O (A5)	82	83	86	K1	17
I/O	–	84	87	H3	20
I/O	–	85	88	J1	23
I/O (A6)	83	86	89	H2	26
I/O (A7)	84	87	90	H1	29
GND	1	88	91	G2	–

* Indicates unconnected package pins.
† Contributes only one bit (.i) to the boundary scan register.
Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 247 = BSCANT.UPD

XC4004A Pinouts

Pin Description	PC84	TQ144	PQ160	PG120	Bound Scan
VCC	2	128	142	G3	–
I/O (A8)	3	129	143	G1	38
I/O (A9)	4	130	144	F1	41
I/O	–	131	145	E1	44
I/O	–	132	146	F2	47
I/O (A10)	5	133	147	F3	50
I/O (A11)	6	134	148	D1	53
–	–	135*	149*	–	–
–	–	136*	150*	–	–
GND	–	137	151	E2	–
–	–	–	152*	–	–
–	–	–	153*	–	–
I/O (A12)	7	138	154	C1	56
I/O (A13)	8	139	155	D2	59
I/O	–	140	156	E3	62
I/O	–	141	157	B1	65
I/O (A14)	9	142	158	C2	68
SGCK1 (A15, I/O)	10	143	159	D3	71
VCC	11	144	160	C3	–
GND	12	1	1	C4	–
PGCK1 (A16, I/O)	13	2	2	B2	74
I/O (A17)	14	3	3	B3	77
I/O	–	4	4	A1	80
I/O	–	5	5	A2	83
I/O (TDI)	15	6	6	C5	86
I/O (TCK)	16	7	7	B4	89
–	–	–	8*	–	–
–	–	–	9*	–	–
GND	–	8	10	A3	–
–	–	9*	11*	–	–
–	–	10*	12*	–	–
I/O (TMS)	17	11	13	B5	92
I/O	18	12	14	A4	95
I/O	–	13	15	C6	98
I/O	–	14	16	A5	101
I/O	19	15	17	B6	104
I/O	20	16	18	A6	107
GND	21	17	19	B7	–
VCC	22	18	20	C7	–
I/O	23	19	21	A7	110
I/O	24	20	22	A8	113
I/O	–	21	23	A9	116
I/O	–	22	24	B8	119
I/O	25	23	25	C8	122
I/O	26	24	26	A10	125
–	–	25*	27*	–	–
–	–	26*	28*	–	–
GND	–	27	29	–	–
–	–	–	30*	–	–
–	–	–	31*	–	–
I/O	27	28	32	B9	128
I/O	–	29	33	A11	131
I/O	–	30	34	B10	134
I/O	–	31	35	–	137

Pin Description	PC84	TQ144	PQ160	PG120	Bound Scan
I/O	28	32	36	C9	140
SGCK2 (I/O)	29	33	37	A12	143
O (M1)	30	34	38	B11	146
GND	31	35	39	C10	–
I (M0)	32	36	40	C11	149†
VCC	33	37	41	D11	–
I (M2)	34	38	42	B12	150†
PGCK2 (I/O)	35	39	43	C12	151
I/O (HDC)	36	40	44	A13	154
I/O	–	41	45	B13	157
I/O	–	42	46	E11	160
I/O	–	43	47	D12	163
I/O ($\overline{LDC}$)	37	44	48	C13	166
–	–	–	49*	–	–
–	–	–	50*	–	–
GND	–	45	51	–	–
–	–	46*	52*	–	–
–	–	47*	53*	–	–
I/O	38	48	54	E12	169
I/O	39	49	55	D13	172
I/O	–	50	56	F11	175
I/O	–	51	57	E13	178
I/O	40	52	58	F12	181
I/O ($\overline{ERR}$, $\overline{INIT}$)	41	53	59	F13	184
VCC	42	54	60	G12	–
GND	43	55	61	G11	–
I/O	44	56	62	G13	187
I/O	45	57	63	H13	190
I/O	–	58	64	J13	193
I/O	–	59	65	H12	196
I/O	46	60	66	H11	199
I/O	47	61	67	K13	202
–	–	62*	68*	–	–
–	–	63*	69*	–	–
GND	–	64	70	–	–
–	–	–	71*	–	–
–	–	–	72*	–	–
I/O	48	65	73	J12	205
I/O	49	66	74	L13	201
I/O	–	67	75	K12	211
I/O	–	68	76	J11	214
I/O	50	69	77	M13	217
SGCK3 (I/O)	51	70	78	L12	220
GND	52	71	79	K11	–
DONE	53	72	80	L11	–
VCC	54	73	81	L10	–
$\overline{PROG}$	55	74	82	M12	–
I/O (D7)	56	75	83	M11	223
PGCK3 (I/O)	57	76	84	N13	226
I/O	-	77	85	N12	229
I/O	-	78	86	L9	232
I/O (D6)	58	79	87	M10	235
I/O	-	80	88	N11	238
-	-	-	89*	-	–

Pin Description	PC84	TQ144	PQ160	PG120	Bound Scan
–	–	–	90*	–	–
GND	–	81	91	–	–
–	–	82*	92*	–	–
–	–	83*	93*	–	–
I/O (D5)	59	84	94	M9	241
I/O ($\overline{CSO}$)	60	85	95	N10	244
I/O	–	86	96	L8	247
I/O	–	87	97	N9	250
I/O (D4)	61	88	98	M8	253
I/O	62	89	99	N8	256
VCC	63	90	100	M7	–
GND	64	91	101	L7	–
I/O (D3)	65	92	102	N7	259
I/O ($\overline{RS}$)	66	93	103	N6	262
I/O	–	94	104	N5	265
I/O	–	95	105	M6	268
I/O (D2)	67	96	106	L6	271
I/O	68	97	107	N4	274
–	–	98*	108*	–	–
–	–	99*	109*	–	–
GND	–	100	110	–	–
–	–	–	111*	–	–
–	–	–	112*	–	–
I/O (D1)	69	101	113	M5	277
I/O (RCLK-$\overline{BUSY}$/RDY)	70	102	114	N3	280
I/O	–	103	115	M4	283
I/O	–	104	116	L5	286
I/O (D0, DIN)	71	105	117	N2	289
SGCK4 (DOUT, I/O)	72	106	118	M3	292
CCLK	73	107	119	L4	–
VCC	74	108	120	L3	–
O (TDO)	75	109	121	M2	–
GND	76	110	122	K3	–
I/O (A0, $\overline{WS}$)	77	111	123	L2	2
PGCK4 (I/O,A1)	78	112	124	N1	5
I/O	–	113	125	M1	8
I/O	–	114	126	J3	11
I/O (CS1, A2)	79	115	127	K2	14
I/O (A3)	80	116	128	L1	17
–	–	117*	129*	–	–
–	–	–	130*	–	–
GND	–	118	131	–	–
–	–	119*	132*	–	–
–	–	120*	133*	–	–
I/O (A4)	81	121	134	J2	20
I/O (A5)	82	122	135	K1	23
–	–	–	136*	–	–
I/O	–	123	137	H3	26
I/O	–	124	138	J1	29
I/O (A6)	83	125	139	H2	32
I/O (A7)	84	126	140	H1	35
GND	1	127	141	G2	–

* Indicates unconnected package pins.
† Contributes only one bit (.i) to the boundary scan register.
Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 295 = BSCANT.UPD

XC4005A Pinouts

Pin Description	PC84	TQ144	PQ160	PQ208	PG156	Bound Scan
VCC	2	128	142	183	H3	–
I/O (A8)	3	129	143	184	H1	44
I/O (A9)	4	130	144	185	G1	47
I/O	–	131	145	186	G2	50
I/O	–	132	146	187	G3	53
–	–	–	–	188*	–	–
–	–	–	–	189*	–	–
I/O (A10)	5	133	147	190	F1	56
I/O (A11)	6	134	148	191	F2	59
I/O	–	135	149	192	E1	62
I/O	–	136	150	193	E2	65
GND	–	137	151	194	F3	–
–	–	–	–	195*	–	–
–	–	–	–	196*	–	–
–	–	–	152*	197*	D1*	–
–	–	–	153*	198*	D2*	–
I/O (A12)	7	138	154	199	E3	68
I/O (A13)	8	139	155	200	C1	71
–	–	–	–	–	–	–
I/O	–	140	156	201	C2	74
I/O	–	141	157	202	D3	77
I/O (A14)	9	142	158	203	B1	80
SGCK1 (A15, I/O)	10	143	159	204	B2	83
VCC	11	144	160	205	C3	–
–	–	–	–	206*	–	–
–	–	–	–	207*	–	–
–	–	–	–	208*	–	–
–	–	–	–	1*	–	–
GND	12	1	1	2	C4	–
–	–	–	–	3*	–	–
PGCK1 (A16, I/O)	13	2	2	4	B3	86
I/O (A17)	14	3	3	5	A1	89
I/O	–	4	4	6	A2	92
I/O	–	5	5	7	C5	95
–	–	–	–	–	–	–
I/O (TDI)	15	6	6	8	B4	98
I/O (TCK)	16	7	7	9	A3	101
–	–	–	8*	10*	A4*	–
–	–	–	9*	11*	–	–
–	–	–	–	12*	–	–
–	–	–	–	13*	–	–
GND	–	8	10	14	C6	–
I/O	–	9	11	15	B5	104
I/O	–	10	12	16	B6	107
I/O (TMS)	17	11	13	17	A5	110
I/O	18	12	14	18	C7	113
–	–	–	–	19*	–	–
–	–	–	–	20*	–	–
I/O	–	13	15	21	B7	116
I/O	–	14	16	22	A6	119
I/O	19	15	17	23	A7	122
I/O	20	16	18	24	A8	125
GND	21	17	19	25	C8	–
VCC	22	18	20	26	B8	–
I/O	23	19	21	27	C9	128
I/O	24	20	22	28	B9	131
I/O	–	21	23	29	A9	134
I/O	–	22	24	30	B10	137
–	–	–	–	31*	–	–
–	–	–	–	32*	–	–
I/O	25	23	25	33	C10	140
I/O	26	24	26	34	A10	143
I/O	–	25	27	35	A11	146
I/O	–	26	28	36	B11	149
GND	–	27	29	37	C11	–
–	–	–	–	38*	–	–
–	–	–	–	39*	–	–
–	–	–	30*	40*	A12*	–
–	–	–	31*	41*	–	–
I/O	27	28	32	42	B12	152
I/O	–	29	33	43	A13	155
I/O	–	30	34	44	A14	158

Pin Description	PC84	TQ144	PQ160	PQ208	PG156	Bound Scan
I/O	–	31	35	45	C12	161
–	–	–	–	–	–	–
I/O	28	32	36	46	B13	164
SGCK2 (I/O)	29	33	37	47	B14	167
O (M1)	30	34	38	48	A15	170
GND	31	35	39	49	C13	–
I (M0)	32	36	40	50	A16	173†
–	–	–	–	51*	–	–
–	–	–	–	52*	–	–
–	–	–	–	53*	–	–
–	–	–	–	54*	–	–
VCC	33	37	41	55	C14	–
I (M2)	34	38	42	56	B15	174†
PGCK2 (I/O)	35	39	43	57	B16	175
I/O (HDC)	36	40	44	58	D14	178
I/O	–	41	45	59	C15	181
–	–	–	–	–	–	–
I/O	–	42	46	60	D15	184
I/O	–	43	47	61	E14	187
I/O (LDC)	37	44	48	62	C16	190
–	–	–	49*	63*	E15*	–
–	–	–	50*	64*	D16*	–
–	–	–	–	65*	–	–
–	–	–	–	66*	–	–
GND	–	45	51	67	F14	–
I/O	–	46	52	68	F15	193
I/O	–	47	53	69	E16	196
I/O	38	48	54	70	F16	199
I/O	39	49	55	71	G14	202
–	–	–	–	72*	–	–
–	–	–	–	73*	–	–
I/O	–	50	56	74	G15	205
I/O	–	51	57	75	G16	208
I/O	40	52	58	76	H16	211
I/O (ERR, INIT)	41	53	59	77	H15	214
VCC	42	54	60	78	H14	–
GND	43	55	61	79	J14	–
I/O	44	56	62	80	J15	217
I/O	45	57	63	81	J16	220
I/O	–	58	64	82	K16	223
I/O	–	59	65	83	K15	226
–	–	–	–	84*	–	–
–	–	–	–	85*	–	–
I/O	46	60	66	86	K14	229
I/O	47	61	67	87	L16	232
I/O	–	62	68	88	M16	235
I/O	–	63	69	89	L15	238
GND	–	64	70	90	L14	–
–	–	–	–	91*	–	–
–	–	–	–	92*	–	–
–	–	–	71*	93*	N16*	–
–	–	–	72*	94*	M15*	–
I/O	48	65	73	95	P16	241
I/O	49	66	74	96	M14	244
I/O	–	67	75	97	N15	247
I/O	–	68	76	98	P15	250
I/O	50	69	77	99	N14	253
SGCK3 (I/O)	51	70	78	100	R16	256
GND	52	71	79	101	P14	–
–	–	–	–	102*	–	–
DONE	53	72	80	103	R15	–
–	–	–	–	104*	–	–
–	–	–	–	105*	–	–
VCC	54	73	81	106	P13	–
–	–	–	–	107*	–	–
PROG	55	74	82	108	R14	–
I/O (D7)	56	75	83	109	T16	259
PGCK3 (I/O)	57	76	84	110	T15	262
I/O	–	77	85	111	R13	265
–	–	–	–	–	–	–
I/O	–	78	86	112	P12	268
I/O(D6)	58	79	87	113	T14	271

* Indicates unconnected package pins.
† Contributes only one bit (.i) to the boundary scan register.

XC4005A Pinouts (continued)

Pin Descriptions	PC84	TQ144	PQ160	PQ208	PG156	Bound Scan
I/O	–	80	88	114	T13	274
–	–	–	89*	115*	R12*	–
–	–	–	90*	116*	T12*	–
–	–	–	–	117*	–	–
–	–	–	–	118*	–	–
GND	–	81	91	119	P11	–
I/O	–	82	92	120	R11	277
I/O	–	83	93	121	T11	280
I/O (D5)	59	84	94	122	T10	283
I/O ($\overline{CS0}$)	60	85	95	123	P10	286
–	–	–	–	124*	–	–
–	–	–	–	125*	–	–
I/O	–	86	96	126	R10	289
I/O	–	87	97	127	T9	292
I/O (D4)	61	88	98	128	R9	295
I/O	62	89	99	129	P9	298
VCC	63	90	100	130	R8	–
GND	64	91	101	131	P8	–
I/O (D3)	65	92	102	132	T8	301
I/O ($\overline{RS}$)	66	93	103	133	T7	304
I/O	–	94	104	134	T6	307
I/O	–	95	105	135	R7	310
–	–	–	–	136*	–	–
–	–	–	–	137*	–	–
I/O (D2)	67	96	106	138	P7	313
I/O	68	97	107	139	T5	316
I/O	–	98	108	140	R6	319
I/O	–	99	109	141	T4	322
GND	–	100	110	142	P6	–
–	–	–	–	143*	–	–
–	–	–	–	144*	–	–
–	–	–	111*	145*	R5*	–
–	–	–	112*	146*	–	–
I/O (D1)	69	101	113	147	T3	325
I/O (RCLK-$\overline{BUSY}$/RDY)	70	102	114	148	P5	328
I/O	–	103	115	149	R4	331
–	–	–	–	–	–	–
I/O	–	104	116	150	R3	334
I/O (D0, DIN)	71	105	117	151	P4	337
SGCK4 (DOUT, I/O)	72	106	118	152	T2	340
CCLK	73	107	119	153	R2	–
VCC	74	108	120	154	P3	–
–	–	–	–	155*	–	–
–	–	–	–	156*	–	–
–	–	–	–	157*	–	–
–	–	–	–	158*	–	–
O (TDO)	75	109	121	159	T1	–
GND	76	110	122	160	N3	–
I/O (A0,$\overline{WS}$)	77	111	123	161	R1	2
PGCK4 (A1,I/O)	78	112	124	162	P2	5
I/O	–	113	125	163	N2	8
–	–	–	–	–	–	–
I/O	–	114	126	164	M3	11
I/O (CS1,A2)	79	115	127	165	P1	14
I/O (A3)	80	116	128	166	N1	17
–	–	117*	129*	167*	M2*	–
–	–	–	130*	168*	M1*	–
–	–	–	–	169*	–	–
–	–	–	–	170*	–	–
GND	–	118	131	171	L3	–
I/O	–	119	132	172	L2	20
I/O	–	120	133	173	L1	23
I/O (A4)	81	121	134	174	K3	26
I/O (A5)	82	122	135	175	K2	29
–	–	–	–	176*	–	–
–	–	–	136*	177*	–	–
I/O	–	123	137	178	K1	32
I/O	–	124	138	179	J1	35
I/O (A6)	83	125	139	180		38
I/O (A7)	84	126	140	181	J3	41
GND	1	127	141	182	H2	–

* Indicates unconnected package pins.
Boundary Scan Bit 0 = TDO.T

Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 343 = BSCANT.UPD

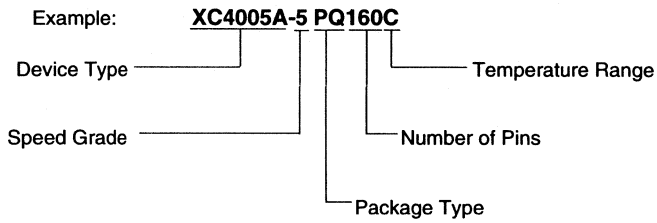
For a detailed description of the device architecture, see pages 2-9 through 2-31.

For a detailed description of the configuration modes and their timing, see pages 2-32 through 2-55.

For detailed lists of package pinouts, see pages 2-57 through 2-81 through 2-85.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS		84	100			120	144	156	160	164	191	196	208		223	240
TYPE		PLAST. PLCC	PLAST. PQFP	PLAST. VQFP	TOP BRAZED CQFP	CERAM. PGA	PLAST. TQFP	CERAM PGA	PLAST. PQFP	TOP BRAZED CQFP	CERAM. PGA	TOP BRAZED CQFP	PLAST. PQFP	METAL PQFP	CERAM. PGA	METAL PQFP
CODE		PC84	PQ100	VQ100	CB100	PG120	TQ144	PG156	PQ160	CB164	PG191	CB196	PQ208	MQ208	PG223	MQ240
XC4002A	-6	C I	C I	C I		C I										
	-5	C	C	C		C										
	-4	C	C	C		C										
XC4003A	-10				M B	M B										
	-6	C I	C I	C I	M B	C I M B										
	-5	C	C	C		C										
	-4	C	C	C		C										
XC4004A	-6	C I				C I	C I		C I							
	-5	C				C	C		C							
	-4	C				C	C		C							
XC4005A	-6	C I					C I	C I	C I				C I			
	-5	C					C	C	C				C			
	-4	C					C	C	C				C			

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C

B = MIL-STD-883C Class B Parentheses indicates future product plans

XC4000H High I/O Count Logic Cell Array Family

Product Specifications

Features

- Third-generation Field-Programmable Gate Arrays
  - Very high number of I/O pins
  - Abundant flip-flops
  - Flexible function generators
  - On-chip ultra-fast RAM
  - Dedicated high-speed carry-propagation circuit
  - Wide edge decoders (four per edge)
  - Efficient implementation of multi-level logic
  - Hierarchy of interconnect lines
  - Internal 3-state bus capability
  - Eight global low-skew clock or signal distribution network
  - IEEE 1149.1-compatible boundary-scan logic support
  - Programmable output slew rate with (two modes including SoftEdge)
- Per-pin individually configurable input threshold and output high level, either TTL or CMOS
  - Programmable input pull-up or pull-down resistors
- Flexible Array Architecture
  - Programmable logic blocks and I/O blocks
  - Programmable interconnects and wide decoders
- Sub-micron CMOS Process
  - High-speed logic and interconnect
  - Low power consumption
- Configured by Loading Binary File
  - Unlimited reprogrammability
  - Six programming modes
- XACT Development System runs on '386/'486-type PC, NEC PC, Apollo, Sun-4, and Hewlett Packard 700 Series
  - Interfaces to popular design environments like Viewlogic, Mentor Graphics and OrCAD
  - Fully automatic partitioning, placement and routing
  - Interactive design editor for design optimization
  - 288 macros, 34 hard macros, RAM/ROM compiler

Description

The XC4000 family of Field-Programmable Gate Arrays (FPGAs) provides the benefits of custom CMOS VLSI, while avoiding the initial cost, time delay, and inherent risk of a conventional masked gate array.

The XC4000 family provides a regular, flexible, programmable architecture of Configurable Logic Blocks (CLBs), interconnected by a powerful hierarchy of versatile routing resources, and surrounded by a perimeter of programmable Input/Output Blocks (IOBs).

Device	XC4003H	XC4005H
Approximate Gate Count	3,000	5,000
Number of IOBs	160	192
CLB Matrix	10 x 10	14 x 14
Number of CLBs	100	196
Number of Flip-Flops	200	392
Max Decode Inputs (per side)	30	42
Max RAM Bits	3,200	6,272

The XC4000H family is intended for I/O-intensive applications. Compared to the XC4000, the XC4000H devices have almost double the number of IOBs and I/O pins, and offer a choice of CMOS- or TTL-level outputs and input thresholds, selectable per pin. The XC4000H outputs sink 24 mA and offer improved 3-state and slew-rate control.

The devices are customized by loading configuration data into the internal memory cells. The FPGA can either actively read configuration data out of external serial or byte-parallel PROM (master modes), or the configuration data can be written into the FPGA (slave and peripheral modes).

The XC4000H family is supported by the same powerful and sophisticated software as the XC4000 family, covering every aspect of design: from schematic entry, to simulation, to automatic block placement and routing of interconnects, and finally to the creation of the configuration bit stream.

Since Xilinx FPGAs can be reprogrammed an unlimited number of times, they can be used in innovative designs where hardware is changed dynamically, or where hardware must be adapted to different user applications. FPGAs are ideal for shortening the design and development cycle, but they also offer a cost-effective solution for production rates well beyond 1000 systems per month.

For a detailed description of the device features, architecture, configuration methods and pin descriptions, see pages 2-9 through 2-45.

XC4000H Compared to XC4000

For readers already familiar with the XC4000 family, here is a concise list of the major new features in the XC4000H family.

- Number of IOBs is, roughly, doubled compared to the XC4000.
- Output slew-rate control is significantly improved.

  Resistive Load means a strong pull-down all the way to ground, capable of sinking 24 mA continuously. If many outputs switch simultaneously, the resulting ground bounce might be objectionable.

  Capacitive Load, or SoftEdge, means a more sophisticated pull-down that decreases in strength as it approaches ground. It can only sink 4 mA at V_{OL}, which is irrelevant when driving capacitive loads. The benefit is a substantial reduction in ground bounce when several outputs switch simultaneously.

  In the XC4000, limiting the slew rate of the output reduces ground bounce, but also introduces a significant additional delay. In the XC4000H, the additional delay in the capacitive-load mode is usually insignificant.
- All input and output flip-flops have been eliminated in the XC4000H family. Use the CLB flip-flops instead.
- Outputs can sink 24 mA, guaranteed at V_{OL} = 0.5 V, compared to the 12 mA at 0.4 V of the XC4000 family.
- Number of decoder inputs per side
- Each output may be individually configured as one of the following.
  - TTL-compatible (like the XC4000) that uses n-channel transistors for both pull-down and pull-up,
  - A totem-pole output structure with reduced V_{OH},
  - CMOS-compatible (like the XC2000 and XC3000) that means n-channel pull-down and p-channel pull-up with V_{OH} close to the V_{CC} rail.
- Each input can individually be configured for either TTL-compatible threshold (1.2 V) or for CMOS-compatible threshold (V_{CC}/2). Each input can be configured to be inverting or non-inverting.
- Any combination of programmable input and output levels on any I/O pin is possible, even the dubious combination of TTL output and CMOS input on the same I/O pin.
- Output 3-state operation is controlled by a two-input multiplexer.
- The first activation of outputs after the end of the configuration process, as they change from 3-state to their active level, is always in the SoftEdge mode. This prevents potential ground-bounce problems when all outputs turn on simulataneously. A few nanoseconds later, each output assumes the current-sink capability determined by its configuration. This soft wake-up operation is transparent to the user.

Architectural Overview

Except for the I/O structure, the XC4000H family is identical to the original XC4000 family. A matrix of Configurable Logic Blocks is interconnected through a hierarchy of flexible routing resources. The powerful system-integration features of the XC4000 family, such as on-chip RAM, dedicated fast carry, and wide decoders, are retained in the XC4000H family.

The XC4000H family almost doubles the number of input/output pins compared to the XC4000, an attractive feature for I/O-intensive applications. The output drivers were redesigned to be more powerful and more flexible.

Input/Output Blocks (IOBs)

The IOBs form the interface between the internal logic and the I/O pads of the XC4000H device. Each IOB consists of a programmable output section that can drive the pad, and a programmable input section, that can receive data from the pad. Aside from being connected to the same pad, the input and output sections have nothing else in common.

Input

In XC4000H devices, there are no input flip-flops.

The input section receives data from the pad. Each input can be configured individually with TTL or CMOS input thresholds. As a configuration option, the input can be either inverted or non-inverted, before it is made available to the internal logic.

Pad

Each I/O pad can be configured with or without a pull-up or pull-down resistor, independent of the pin usage.

Boundary Scan

The XC4000H IOBs have the same IEE 1149.1boundary-scan capabilities as the IOBs in the original XC4000.

Output

In an XC4000H IOB, there is no output flip-flop. The output section receives data and 3-state control information from the CLB interconnect structure.

Under configuration control, the data can be inverted or non-inverted. The output driver assumes one of the following states.

- Permanently disabled, making the pad an input only pad
- 3-state controlled from the internal logic

There are two potential sources of the 3-state-control information, selected by a multiplexer. The output of the multiplexer driving the 3-state control can be inverted as a configuration option. The signal can be active High 3-state, which is identical to the more popular connotation of active-Low Output Enable, or it can be active-High Output Enable, which is identical to active Low 3-state.

Each output can be individually configured as either TTL- or CMOS-compatible. A TTL-compatible output uses n-channel transistors for both pull-down and pull-up. As a result, the output High voltage, V_{OH}, is at least one threshold voltage drop below V_{CC}. Depending on the load current, this means a voltage drop of 1.0 to 2.4 V. In a system using TTL input thresholds of 1.2 V, this lower output voltage results in shorter delays when switching from High to Low, and thus a better delay balance between the two signal directions. The smaller signal amplitude also generates less noise. The reduction in High-level noise margin is irrelevant because it is still much better than the Low-level noise margin. TTL-level outputs are, therefore, the best choice for systems that use TTL-level input thresholds. (XC4000 and XC4000A devices have only TTL-level outputs and have only TTL-level input thresholds).

When the output is configured as CMOS-compatible, an additional p-channel transistor pulls the output towards the V_{CC} rail. This results in an unloaded rail-to-rail signal swing, ideal for systems that use CMOS input thresholds. (XC2000 and XC3000 devices have only CMOS-level outputs).

Each output can be configured for either of two slew-rate options, which affect only the pull-down operation. When configured for resistive load, the pull-down transistor is driven hard, resulting in a practically constant on-resistance of about 10 Ω. This results in the fastest High-to-Low transition, and the capability to sink 24 mA with a voltage of 500 mV. When many outputs switch High to Low simultaneously, especially when they are discharging a capacitive load, this configuration option might result in excessive ground bounce.

When configured for capacitive load, or SoftEdge, the High-to-Low transition starts as described above, but the drive to the pull-down transistor is reduced as soon as the output voltage reaches a value around 1 V. This results in a higher resistance in the pull-down transistor, a slowing down of the falling edge, and a significantly reduced ground bounce.

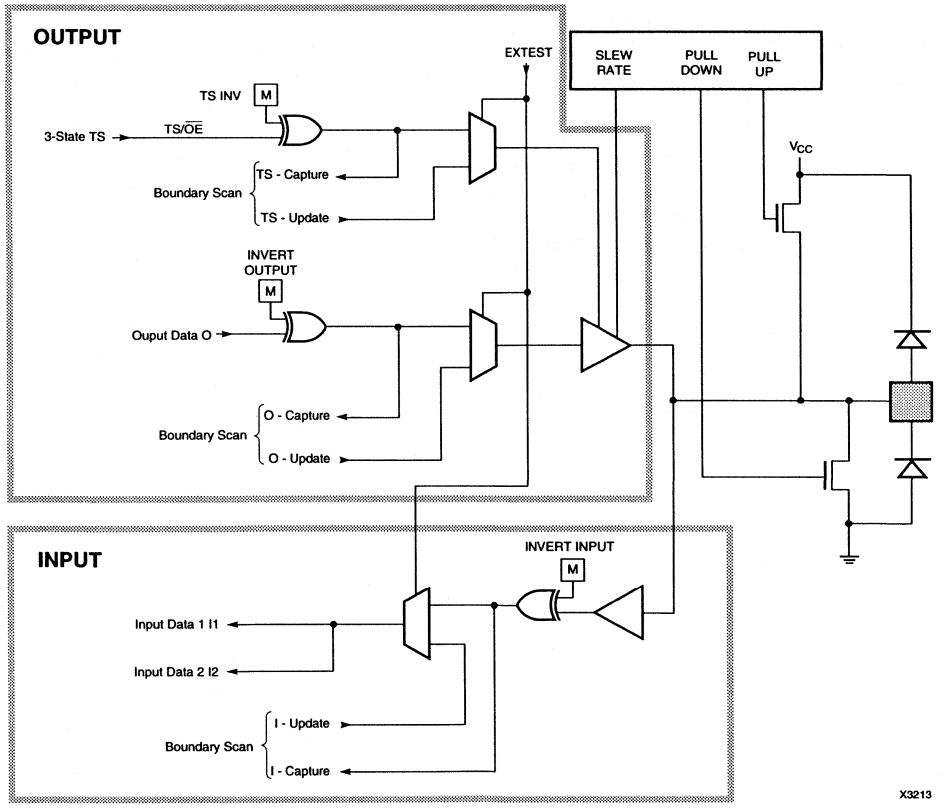


Figure 1. XC4000H Input/Output Block

Slew-Rate Control with SoftEdge

The XC4000H outputs use a novel, patent-pending method of slew-rate control that reduces ground bounce without any significant delay penalty. Each output is configured with a choice between two slew-rate options. Both options reduce the positive ground bounce that occurs when the output current is turned on. They differ in the way the output current is turned off.

- The slew-rate-limited default mode is called capacitive, or SoftEdge. At the beginning of a High-to-Low transition, the pull-down transistor is gradually turned on, and kept fully conductive until the output voltage has reached +1 V. The pull-down transistor is then gradually turned off, so that it finally has an on-resistance of about 100 Ω, low enough to sink 4 mA continuously. Gradually turning off the sink current reduces the max value of current change (di/dt) that is normally responsible for the negative voltage spike over the common ground inductance (bonding wires), called ground bounce.

  The capacitive, or SoftEdge, mode is the best choice for capacitively loaded outputs, or for outputs requiring less than 4 mA of dc sink current.

- The non-slew-rate limited mode is called resistive. At the beginning of a High-to-Low transition, the pull-down transistor is gradually turned on, and kept fully conductive as long as the output data is a logic Low. The pull-down transistor has an impedance of <20 Ω, capable of sinking 24 mA continuously.

  Resisitive mode is required for driving terminated transmission lines with 4 to 24 mA of dc sink current. The abrupt current change when the output voltage reaches zero causes a voltage spike over the ground inductance (bonding wire) and can result in objectionable ground bounce when many outputs switch High-to-Low simultaneously.

The following figures show output rising and falling edges when one output drives different loads. The tests were performed on a multi-ground-plane test PC board, manufactured by Urban Instruments (Encino, CA). Measurements were done with a Tektronix TDS540 digital storage oscilloscope. The figures below are unedited files from these measurements, the time scale is 2 ns/division.

The upper trace in each figure shows a second output driven from the same internal signal, but unloaded. It acts as a timing reference, and triggers the oscilloscope.

Resistive mode and capacitive mode transitions start with practically the same delay from the internal logic. Resisitive mode falls faster, and has more undershoot; capacitive mode rises slightly faster. For a 200-Ω pull-up, 330-Ω pull-down termination, only resisitive mode is meaningful. A TTL-output with a 1000-Ω pull-up, 150-pF termination has a slow (150 ns) final rise time that extends outside the 10-ns timing window of these figures.

Trace A shows Resistive mode with CMOS outputs
Trace B shows Resistive mode with TTL outputs
Trace C shows Capacitive mode with CMOS outputs
Trace D shows Capacitive mode with TTL outputs

Summary

Use resistive mode for applications that require >4 mA of dc sink current, and for heavy capacitive loads when they must be discharged fast. Use capacitive mode for all other applications, especially for light capacitive loads (50 to 200 pF) and for all timing-uncritical outputs that require <4 mA dc current. The Low-to-High transition is not affected by the choice of slew-rate mode.

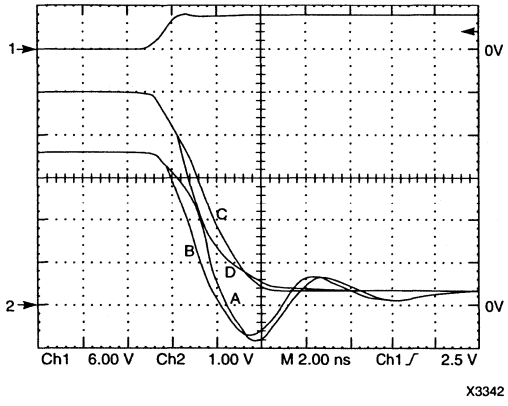


Figure 2. Falling Edge, 50 pF Load

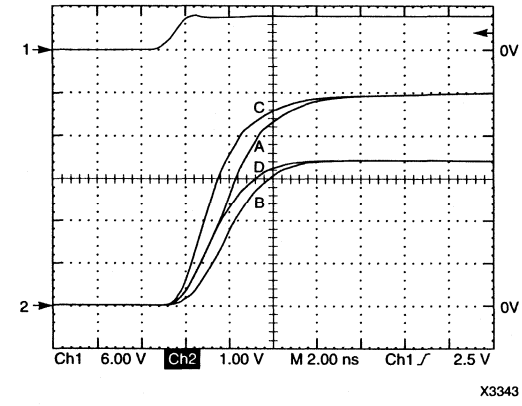


Figure 3. Rising Edge, 50 pF Load

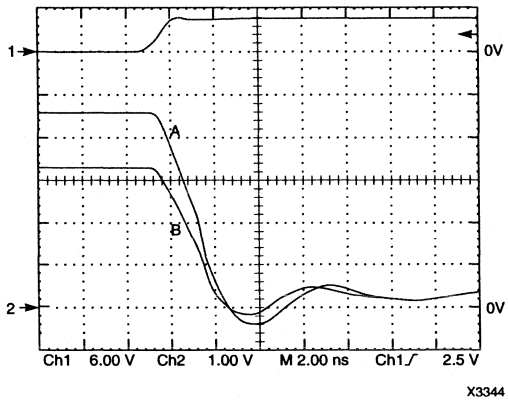


Figure 4. Falling Edge, 200/330 Ω, 50 pF Load

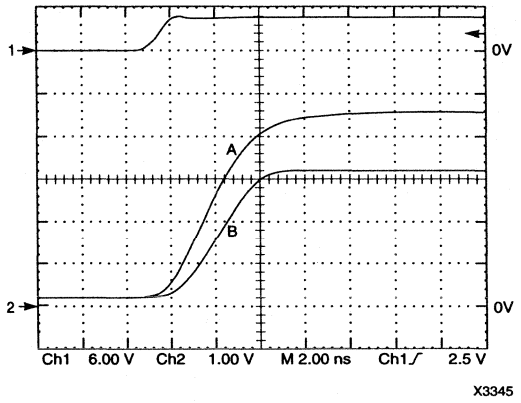


Figure 5. Rising Edge, 200/330 Ω, 50 pF Load

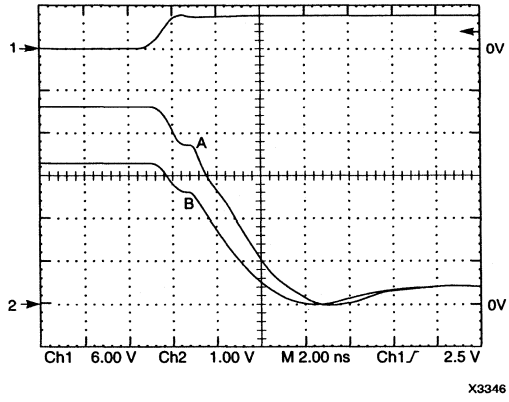


Figure 6. Falling Edge, 200/330 Ω, 150 pF Load

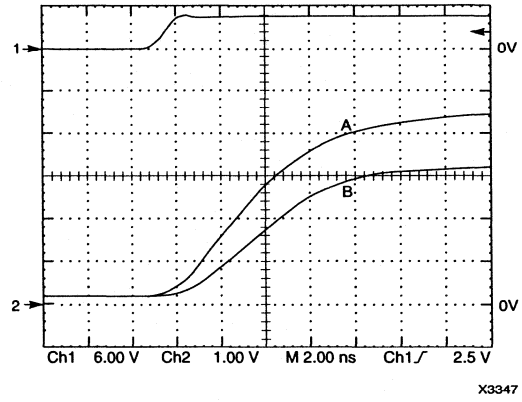


Figure 7. Rising Edge, 200/330 Ω, 150 pF Load

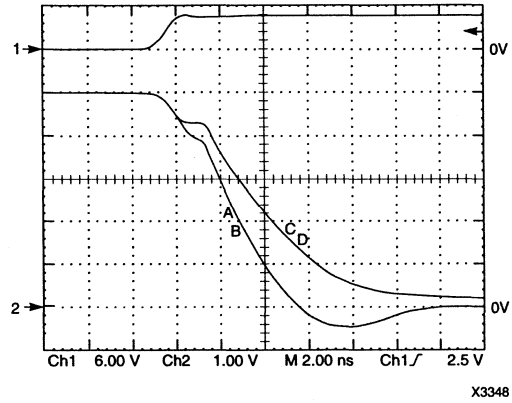


Figure 8. Falling Edge, 1000 Ω, 150 pF Load

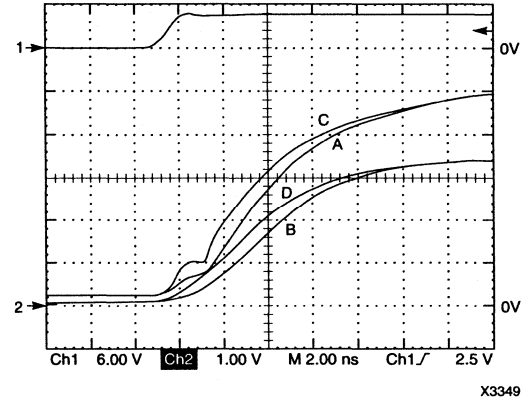


Figure 9. Rising Edge, 1000 Ω, 150 pF Load

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to 7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to 7.0	V
V_{TS}	Voltage applied to 3-state output	–0.5 to 7.0	V
T_{STG}	Storage temperature (ambient)	–65 to + 150	°C
T_J	Junction temperature	+ 150	°C

Note: Stresses beyond those listed under absolute maximum ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description			Min	Max	Units
V_{CC}	Supply voltage relative to GND	Commercial	0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND	Industrial	–40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND	Military	–55°C to 125°C	4.5	5.5	V
V_{IH}	High-level input voltage for TTL threshold			2.0	V_{CC}	V
V_{IH}	High-level input voltage for CMOS threshold			70%	100%	V_{CC}
V_{IL}	Low-level input voltage for TTL threshold			0	0.8	V
V_{IL}	Low-level input voltage CMOS threshold			0	20%	V_{CC}

DC Characteristics Over Operating Conditions

Symbol	Description	Min	Max	Units
V_{OH}	High-level output voltage, TTL option @ I_{OH} = –4.0 mA	2.4		V
V_{OH}	High-level output voltage, CMOS option @ I_{OH} = -1 mA		V_{CC} -0.5	V
V_{OL}	Low-level output voltage @ I_{OL} = 24 mA, V_{CC} max (Note 1)		0.5	V
I_{CCO}	Quiescent LCA supply current (Note 2)		10	mA
I_{IL}	Leakage current	–10	+10	µA
C_{IN}	Input capacitance (sample tested)		15	pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0V (estimate)	0.02	0.20	mA
I_{RLL}	Horizontal Long Line pull-up (when selected) @ logic Low	0.2	2.5	mA

Note: 1. XC4003H–with 50% of the outputs simultaneously sinking 24 mA. XC4005H–with 33% of the outputs simultaneously sinking 24 mA.
2. With no output current loads, no active input or long line pull-resistors, all package pins at V_{CC} or GND, and the LCA configured with a MakeBits tie option.

Wide Decoder Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
Full length, both pull-ups, inputs from IOB i-pins	T_{WAF}	XC4003H XC4005H	9.0 10.0	8.0 9.0	5.0 6.0	ns ns
Full length, both pull-ups inputs from internal logic	T_{WAFL}	XC4003H XC4005H	12.0 13.0	11.0 12.0	7.0 8.0	ns ns
Half length, one pull-up inputs from IOB i-pins	T_{WAO}	XC4003H XC4005H	9.0 10.0	8.0 9.0	6.0 7.0	ns ns
Half length, one pull-up inputs from internal logic	T_{WAOL}	XC4003H XC4005H	12.0 13.0	11.0 12.0	8.0 9.0	ns ns

Note: These delays are specified from the decoder input to the decoder output. For pin-to-pin delays, add the input delay (T_{PID}) and output delay (T_{OPR} or T_{OPC}), as listed on page 2-93.

Global Buffer Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
Global Signal Distribution	T_{PG}	XC4003H	7.8	5.8	5.1	ns
From pad through **primary** buffer, to any clock k		XC4005H	8.0	6.0	5.5	ns
From pad through **secondary** buffer, to any clock k	T_{SG}	XC4003H	8.8	6.8	6.3	ns
		XC4005H	9.0	7.0	6.7	ns

Horizontal Longline Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

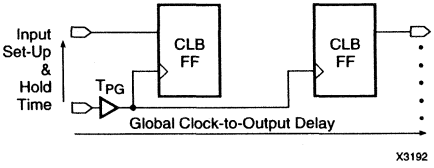
	Speed Grade		-6	-5	-4	
Description	**Symbol**	**Device**	**Max**	**Max**	**Max**	**Units**
TBUF driving a Horizontal Longline (L.L.) I going High or Low to L.L. while T is Low, i.e. buffer is constantly active	T_{IO1}	XC4003H XC4005H	8.8 10.0	6.2 7.0	4.4 5.5	ns ns
I going Low to L.L. going from resistive pull-up High to active Low, (TBUF configured as open drain)	T_{IO2}	XC4003H XC4005H	9.3 10.5	6.7 7.5	5.0 6.0	ns ns
T going Low to L.L. going from resistive pull-up or floating High to active Low, (TBUF configured as open drain)	T_{ON}	XC4003H XC4005H	10.7 12.0	9.0 10.0	7.2 8.0	ns ns
T going High to TBUF going inactive, not driving the L.L.	T_{OFF}	All devices	3.0	2.0	1.8	ns
T going High to L.L. going from Low to High, pulled up by single resistor	T_{PUS}	XC4003H XC4005H	24.0 26.0	20.0 22.0	14.0 16.0	ns ns
T going High to L.L. going from Low to High, pulled up by two resistors	T_{PUF}	XC4003H XC4005H	11.0 12.0	9.0 10.0	7.0 8.0	ns ns

Input and Output Parameters (Pin-to-Pin)

All values listed below are tested directly and guaranteed over the operating conditions. The same parameters can also be derived indirectly from the IOB and Global Buffer specifications. The XACT delay calculator uses this indirect method. When there is a discrepancy between these two methods, the directly tested values listed below should be used, and the indirectly derived values must be ignored.

	Speed Grade		-6	-5	-4*	
Description	**Symbol**	**Device**				**Units**
Global Clock to Output (fast) using nearest CLB FF	T_{ICKOF} (Max)	XC4003H	18.0	15.0		ns
		XC4005H	19.0	16.0		ns
Global Clock to Output (slew limited) using nearest CLB FF	T_{ICKO} (Max)	XC4003H	29.0	24.0		ns
		XC4005H	30.0	25.0		ns
Input Set-up Time, using nearest CLB FF	T_{PSUF} (Min)	XC4003H	5.0	4.0		ns
		XC4005H	4.5	3.5		ns
Input Hold time, using nearest CLB FF	T_{PHF} (Min)	XC4003H	0	0		ns
		XC4005H	0.5	0.5		ns

* Data not available at press time



Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture).

When testing fast outputs, only one output switches. When testing slew-rate limited outputs, half the number of outputs on one side of the device are switching.

These parameter values are tested and guaranteed for worst-case conditions of supply voltage and temperature, and also with the most unfavorable clock polarity choice. The use of a rising-edge clock reduces the effective clock delay by 1 to 2 ns.

The use of a rising clock edge, therefore, reduces the clock-to-output delay, and ends the hold-time requirement earlier. The use of a falling clock edge reduces the input set-up time requirement.

In the tradition of guaranteeing absolute worst-case parameter values, the table above does not take advantage of these improvements. The user can chose between a rising clock edge with slightly shorter output delay, or a falling clock edge with slightly shorter input set-up time. One of these parameters is inevitably better than the guaranteed specification listed above, albeit by only one to two nanoseconds.

CLB Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade	-6		-5		-4		
Description	**Symbol**	**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Combinatorial Delays								
F/G inputs to X/Y outputs	T_{ILO}		6.0		4.5		4.0	ns
F/G inputs via H' to X/Y outputs	T_{IHO}		8.0		7.0		6.0	ns
C inputs via H' to X/Y outputs	T_{HHO}		7.0		5.0		4.5	ns
CLB Fast Carry Logic								
Operand inputs (F1,F2,G1,G4) to C_{OUT}	T_{OPCY}		7.0		5.5		5.0	ns
Add/Subtract input (F3) to C_{OUT}	T_{ASCY}		8.0		6.0		5.5	ns
Initialization inputs (F1,F3) to C_{OUT}	T_{INCY}		6.0		4.0		3.5	ns
C_{IN} through function generators to X/Y outputs	T_{SUM}		8.0		6.0		5.5	ns
C_{IN} to C_{OUT}, bypass function generators.	T_{BYP}		2.0		1.5		1.5	ns
Sequential Delays								
Clock K to outputs Q	T_{CKO}		5.0		3.0		3.0	ns
Set-up Time before Clock K								
F/G inputs	T_{ICK}	6.0		4.5		4.5		ns
F/G inputs via H'	T_{IHCK}	8.0		6.0		6.0		ns
C inputs via H1	T_{HHCK}	7.0		5.0		5.0		ns
C inputs via DIN	T_{DICK}	4.0		3.0		3.0		ns
C inputs via EC	T_{ECCK}	7.0		4.0		3.0		ns
C inputs via S/R, going Low (inactive)	T_{RCK}	6.0		4.5		4.0		ns
C_{IN} input via F'/G'		8.0		6.0		5.5		ns
C_{IN} input via F'/G' and H'		10.0		7.5		7.3		ns
Hold Time after Clock K								
F/G inputs	T_{CKI}	0		0		0		ns
F/G inputs via H'	T_{CKIH}	0		0		0		ns
C inputs via H1	T_{CKHH}	0		0		0		ns
C inputs via DIN	T_{CKDI}	0		0		0		ns
C inputs via EC	T_{CKEC}	0		0		0		ns
C inputs via S/R, going Low (inactive)	T_{CKR}	0		0		0		ns
Clock								
Clock High time	T_{CH}	5.0		4.0		4.0		ns
Clock Low time	T_{CL}	5.0		4.0		4.0		ns
Set/Reset Direct								
Width (High)	T_{RPW}	5.0		4.0		4.0		ns
Delay from C inputs via S/R, going High to Q	T_{RIO}		9.0		8.0		7.0	ns
Master Set/Reset*								
Width (High or Low)	T_{MRW}	21.0		18.0		18.0		ns
Delay from Global Set/Reset net to Q	T_{MRQ}		33.0		31.0		28.0	ns

* Timing is based on the XC4005H. For other devices see XACT timing calculator.

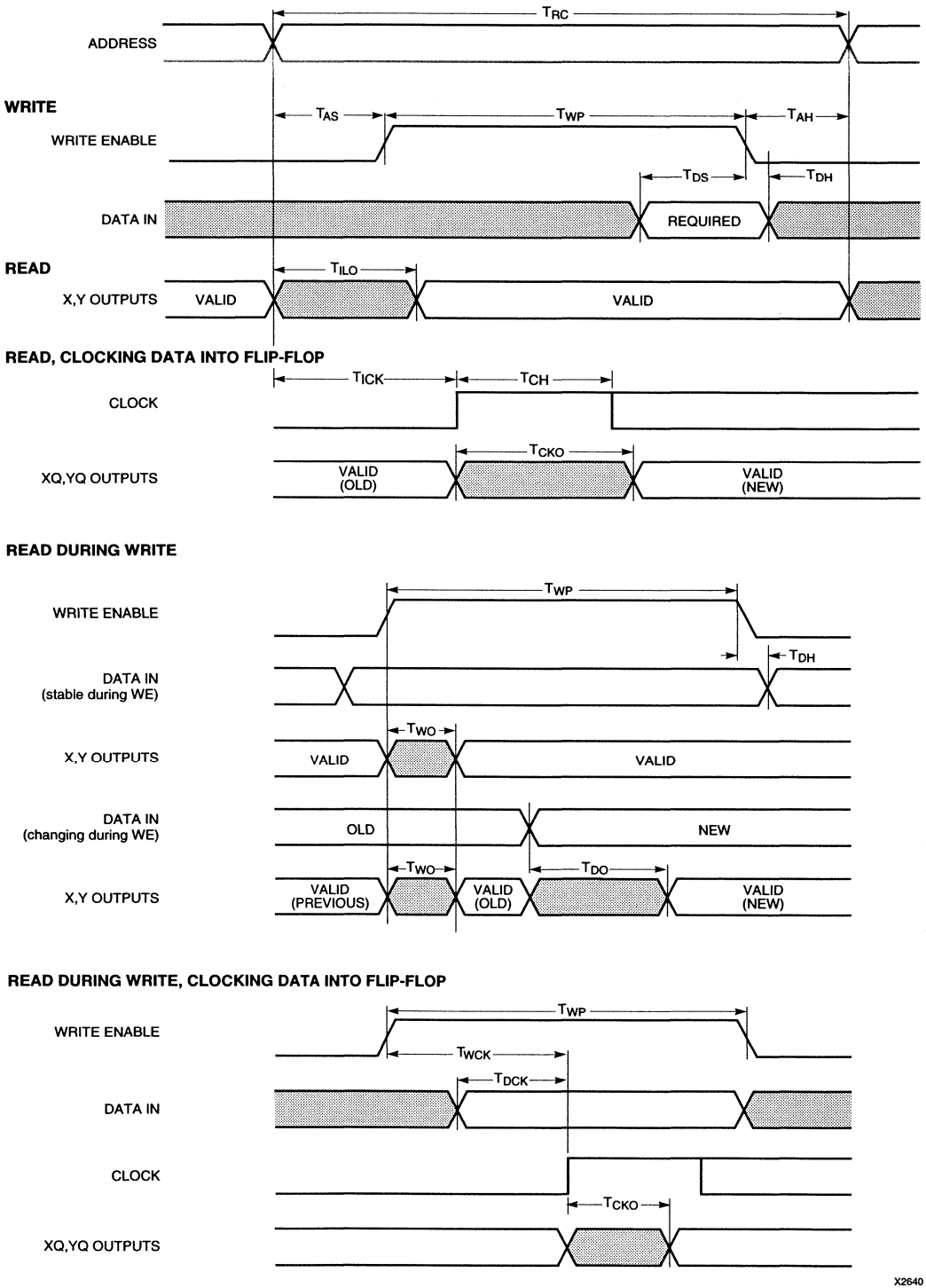
CLB Switching Characteristic Guidelines (continued)

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

CLB RAM Option	**Speed Grade**		**-6**		**-5**		**-4**		
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Write Operation									
Address write cycle time	16 x 2	T_{WC}	9.0		8.0		8.0		ns
	32 x 1	T_{WCT}	9.0		8.0		8.0		ns
Write Enable pulse width (High)	16 x 2	T_{WP}	5.0		4.0		4.0		ns
	32 x 1	T_{WPT}	5.0		4.0		4.0		ns
Address set-up time before beginning of WE	16 x 2	T_{AS}	2.0		2.0		2.0		ns
	32 x 1	T_{AST}	2.0		2.0		2.0		ns
Address hold time after end of WE	16 x 2	T_{AH}	2.0		2.0		2.0		ns
	32 x 1	T_{AHT}	2.0		2.0		2.0		ns
DIN set-up time before end of WE	16 x 2	T_{DS}	4.0		4.0		4.0		ns
	32 x 1	T_{DST}	5.0		5.0		5.0		ns
DIN hold time after end of WE	both	T_{DHT}	2.0		2.0		2.0		ns
Read Operation									
Address read cycle time	16 x 2	T_{RC}	7.0		5.5		5.0		ns
	32 x 1	T_{RCT}	10.0		7.5		7.0		ns
Data valid after address change	16 x 2	T_{ILO}		6.0		4.5		4.0	ns
(no Write Enable)	32 x 1	T_{IHO}		8.0		7.0		6.0	ns
Read Operation, Clocking Data into Flip-Flop									
Address setup time before clock K	16 x 2	T_{ICK}	6.0		4.5		4.5		ns
	32 x 1	T_{IHCK}	8.0		6.0		6.0		ns
Read During Write									
Data valid after WE going active	16 x 2	T_{WO}		12.0		10.0		9.0	ns
(DIN stable before WE)	32 x 1	T_{WOT}		15.0		12.0		11.0	ns
Data valid after DIN	16 x 2	T_{DO}		11.0		9.0		8.5	ns
(DIN change during WE)	32 x 1	T_{DOT}		14.0		11.0		11.0	ns
Read During Write, Clocking Data into Flip-Flop									
WE setup time before clock K	16 x 2	T_{WCK}	12.0		10.0		9.5		ns
	32 x 1	T_{WCKT}	15.0		12.0		11.5		ns
Data setup time before clock K	16 x 2	T_{DCK}	11.0		9.0		9.0		ns
	32 x 1	T_{DCKT}	14.0		11.0		11.0		ns

Note: Timing for the 16 x 1 RAM option is identical to 16 x 2 RAM timing

CLB RAM Timing Characteristics



IOB Switching Characteristic Guidelines

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

Inputs

Description	Symbol	-6 Min	-6 Max	-5 Min	-5 Max	-4 Min	-4 Max	Units
Propagation Delays from CMOS or TTL Levels Pad to I1, I2	T_{PID}		4.0		3.0		2.8	ns

Outputs

Description	Symbol	-6 Min	-6 Max	-5 Min	-5 Max	-4 Min	-4 Max	Units
Propagation Delays to TTL Levels								
Output (O) to Pad (Resistive Mode)	T_{OPR}		9.5		7.5		6.0	ns
Otuput (O) to Pad (Capacitive Mode)	T_{OPC}		10.5		8.0		6.5	ns
3-state to Pad begin hi-Z (Resistive Mode)	T_{TSHZR}		10.5		8.5		7.0	ns
3-state to Pad begin hi-Z (Capacitive Mode)	T_{TSHZC}		8.0		6.5		5.5	ns
3-state to Pad active and valid (Resistive Mode)	T_{TSONR}		14.0		11.0		10.0	ns
3-state to Pad active and valid (Capacitive Mode)	T_{TSONC}		16.0		12.0		11.0	ns
Propagation Delays to CMOS Levels								
Output (O) to Pad (Resistive Mode)	T_{OPR}		9.5		7.5		6.0	ns
Otuput (O) to Pad (Capacitive Mode)	T_{OPC}		9.0		7.0		5.5	ns
3-state to Pad begin hi-Z (Resistive Mode)	T_{TSHZR}		10.5		8.5		7.0	ns
3-state to Pad begin hi-Z (Capacitive Mode)	T_{TSHZC}		8.0		6.5		5.5	ns
3-state to Pad active and valid (Resistive Mode)	T_{TSONR}		14.0		11.0		10.0	ns
3-state to Pad active and valid (Capacitive Mode)	T_{TSONC}		14.0		11.0		10.0	ns

Notes: 1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture).

2. Output delays change with capacitive loading as described in the following table.

	TTL Levels	CMOS Levels	Units
Resistive Mode	0.03	0.03	ns/pF
Capacitive Mode	0.04	0.03	ns/pF

3. Voltage levels of unused (bonded and unbonded) pads must be valid logic levels. Each can be configured with the internal pull-up or pull-down resistor, or alternatively, configured as a driven output or be driven from an external source.

XC4003H Pinouts

Pin Description	PG191	PQ208	Bound Scan
VCC	J4	183	–
I/O (A8)	J3	184	62
I/O (A9)	J2	185	65
I/O	J1	186	68
I/O	H1	187	71
I/O	H2	188	74
I/O	H3	189	77
I/O (A10)	G1	190	80
I/O (A11)	G2	191	83
I/O	F1	192	86
I/O	E1	193	89
GND	G3	194	–
I/O	F2	195	92
I/O	D1	196	95
I/O	C1	197	98
I/O	E2	198	101
I/O (A12)	F3	199	104
I/O (A13)	D2	200	107
I/O	B1	201	110
I/O	E3	202	113
I/O (A14)	C2	203	116
SGCK1 (A15, I/O)	B2	204	119
VCC	D3	205	–
–	–	206*	–
–	–	207*	–
–	–	208*	–
–	–	1*	–
GND	D4	2	–
–	–	3*	–
PGCK1 (A16, I/O)	C3	4	122
I/O (A17)	C4	5	125
I/O	B3	6	128
I/O	C5	7	131
I/O (TDI)	A2	8	134
I/O (TCK)	B4	9	137
I/O	C6	10	140
I/O	A3	11	143
I/O	B5	12	146
I/O	B6	13	149
GND	C7	14	–
I/O	A4	15	152
I/O	A5	16	155
I/O (TMS)	B7	17	158
I/O	A6	18	161
I/O	C8	19	164
I/O	A7	20	167
I/O	B8	21	170
I/O	A8	22	173
I/O	B9	23	176
I/O	C9	24	179
GND	D9	25	–
VCC	D10	26	–

Pin Description	PG191	PQ208	Bound Scan
I/O	C10	27	182
I/O	B10	28	185
I/O	A9	29	188
I/O	A10	30	191
I/O	A11	31	194
I/O	C11	32	197
I/O	B11	33	200
I/O	A12	34	203
I/O	B12	35	206
I/O	A13	36	209
GND	C12	37	–
I/O	B13	38	212
I/O	A14	39	215
I/O	A15	40	218
I/O	C13	41	221
I/O	B14	42	224
I/O	A16	43	227
I/O	B15	44	230
I/O	C14	45	233
I/O	A17	46	236
SGCK2 (I/O)	B16	47	239
O (M1)	C15	48	242
GND	D15	49	–
I (M0)	A18	50	245†
–	–	51*	–
–	–	52*	–
–	–	53*	–
–	–	54*	–
VCC	D16	55	–
I (M2)	C16	56	246†
PGCK2 (I/O)	B17	57	247
I/O (HDC)	E16	58	250
I/O	C17	59	253
I/O	D17	60	256
I/O	B18	61	259
I/O (LDC)	E17	62	262
I/O	F16	63	265
I/O	C18	64	268
I/O	D18	65	271
I/O	F17	66	274
GND	G16	67	–
I/O	E18	68	277
I/O	F18	69	280
I/O	G17	70	283
I/O	G18	71	286
I/O	H16	72	289
I/O	H17	73	292
I/O	H18	74	295
I/O	J18	75	298
I/O	J17	76	301
I/O ($\overline{ERR}$, $\overline{INIT}$)	J16	77	304
VCC	J15	78	–

Pin Description	PG191	PQ208	Bound Scan
GND	K15	79	–
I/O	K16	80	307
I/O	K17	81	310
I/O	K18	82	313
I/O	L18	83	316
I/O	L17	84	319
I/O	L16	85	322
I/O	M18	86	325
I/O	M17	87	328
I/O	N18	88	331
I/O	P18	89	334
GND	M16	90	–
I/O	N17	91	337
I/O	R18	92	340
I/O	T18	93	343
I/O	P17	94	346
I/O	N16	95	349
I/O	T17	96	352
I/O	R17	97	355
I/O	P16	98	358
I/O	U18	99	361
SGCK3 (I/O)	T16	100	364
GND	R16	101	–
–	–	102*	–
DONE	U17	103	–
–	–	104*	–
–	–	105*	–
VCC	R15	106	–
–	–	107*	–
$\overline{PROG}$	V18	108	–
I/O (D7)	T15	109	367
PGCK3 (I/O)	U16	110	370
I/O	T14	111	373
I/O	U15	112	376
I/O (D6)	V17	113	379
I/O	V16	114	382
I/O	T13	115	385
I/O	U14	116	388
I/O	V15	117	391
I/O	V14	118	394
GND	T12	119	–
I/O	U13	120	397
I/O	V13	121	400
I/O (D5)	U12	122	403
I/O (CS0)	V12	123	406
I/O	T11	124	409
I/O	U11	125	412
I/O	V11	126	415
I/O	V10	127	418
I/O (D4)	U10	128	421
I/O	T10	129	424
VCC	R10	130	–

Pin Description	PG191	PQ208	Bound Scan
GND	R9	131	–
I/O (D3)	T9	132	427
I/O ($\overline{RS}$)	U9	133	430
I/O	V9	134	433
I/O	V8	135	436
I/O	U8	136	439
I/O	T8	137	442
I/O (D2)	V7	138	445
I/O	U7	139	448
I/O	V6	140	451
I/O	U6	141	454
GND	T7	142	–
I/O	V5	143	457
I/O	V4	144	460
I/O	U5	145	463
I/O	T6	146	466
I/O (D1)	V3	147	469
I/O (RCLK-$\overline{BUSY}$/RDY)	V2	148	472
I/O	U4	149	475
I/O	T5	150	478
I/O (D0, DIN)	U3	151	481
SGCK4 (DOUT, I/O)	T4	152	484
CCLK	V1	153	–
VCC	R4	154	–
–	–	155*	–
–	–	156*	–
–	–	157*	–
–	–	158*	–
O (TDO)	U2	159	–
GND	R3	160	–
I/O (A0, $\overline{WS}$)	T3	161	2
PGCK4 (I/O, A1)	U1	162	5
I/O	P3	163	8
I/O	R2	164	11
I/O (CS1, A2)	T2	165	14
I/O (A3)	N3	166	17
I/O	P2	167	20
I/O	T1	168	23
I/O	R1	169	26
I/O	N2	170	29
GND	M3	171	–
I/O	P1	172	32
I/O	N1	173	35
I/O (A4)	M2	174	38
I/O (A5)	M1	175	41
I/O	L3	176	44
I/O	L2	177	47
I/O	L1	178	50
I/O	K1	179	53
I/O (A6)	K2	180	56
I/O (A7)	K3	181	59
GND	K4	182	–

* Indicates unconected package pins.
† Contributes only one bit (.i) to the boundary scan register.
Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 487 = BSCANT.UPD

XC4005H Pinouts

Pin Description	PG223	MQ240	Bound Scan
VCC	J4	212	–
I/O (A8)	J3	213	74
I/O (A9)	J2	214	77
I/O	J1	215	80
I/O	H1	216	83
I/O	H2	217	86
I/O	H3	218	89
GND	–	219	–
I/O (A10)	G1	220	92
I/O (A11)	G2	221	95
VCC	–	222	–
I/O	H4	223	98
I/O	G4	224	101
I/O	F1	225	104
I/O	E1	226	107
GND	G3	227	–
I/O	F2	228	110
I/O	D1	229	113
I/O	C1	230	116
I/O	E2	231	119
I/O (A12)	F3	232	122
I/O (A13)	D2	233	125
I/O	F4	234	128
I/O	E4	235	131
I/O	B1	236	134
I/O	E3	237	137
I/O (A14)	C2	238	140
SGCK1 (A15, I/O)	B2	239	143
VCC	D3	240	–
GND	D4	1	–
PGCK1 (A16,I/O)	C3	2	146
I/O (A17)	C4	3	149
I/O	B3	4	152
I/O	C5	5	155
I/O (TDI)	A2	6	158
I/O (TCK)	B4	7	161
I/O	C6	8	164
I/O	A3	9	167
I/O	B5	10	170
I/O	B6	11	173
I/O	D5	12	176
I/O	D6	13	179
GND	C7	14	–
I/O	A4	15	182
I/O	A5	16	185
I/O (TMS)	B7	17	188
I/O	A6	18	191
VCC	–	19	–
I/O	D7	20	194
I/O	D8	21	197
GND	–	22	–
I/O	C8	23	200
I/O	A7	24	203
I/O	B8	25	206
I/O	A8	26	209
I/O	B9	27	212
I/O	C9	28	215
GND	D9	29	–
VCC	D10	30	–
I/O	C10	31	218

Pin Description	PG223	MQ240	Bound Scan
I/O	B10	32	221
I/O	A9	33	224
I/O	A10	34	227
I/O	A11	35	230
I/O	C11	36	233
GND		37	
I/O	D11	38	236
I/O	D12	39	239
VCC	–	40	–
I/O	B11	41	242
I/O	A12	42	245
I/O	B12	43	248
I/O	A13	44	251
GND	C12	45	–
I/O	D13	46	254
I/O	D14	47	257
I/O	B13	48	260
I/O	A14	49	263
I/O	A15	50	266
I/O	C13	51	269
I/O	B14	52	272
I/O	A16	53	275
I/O	B15	54	278
I/O	C14	55	281
I/O	A17	56	284
SGCK2 (I/O)	B16	57	287
O (M1)	C15	58	290
GND	D15	59	–
I (M0)	A18	60	293†
VCC	D16	61	–
I (M2)	C16	62	294†
PGCK2 (I/O)	B17	63	295
I/O (HDC)	E16	64	298
I/O	C17	65	301
I/O	D17	66	304
I/O	B18	67	307
I/O (LDC)	E17	68	310
I/O	F16	69	313
I/O	C18	70	316
I/O	D18	71	319
I/O	F17	72	322
I/O	E15	73	325
I/O	F15	74	328
GND	G16	75	–
I/O	E18	76	331
I/O	F18	77	334
I/O	G17	78	337
I/O	G18	79	340
VCC	–	80	–
I/O	H16	81	343
I/O	H17	82	346
GND	–	83	–
I/O	G15	84	349
I/O	H15	85	352
I/O	H18	86	355
I/O	J18	87	358
I/O	J17	88	361
I/O ($\overline{ERR}$, $\overline{INIT}$)	J16	89	364
VCC	J15	90	–
GND	K15	91	–

Pin Description	PG223	MQ240	Bound Scan
I/O	K16	92	367
I/O	K17	93	370
I/O	K18	94	373
I/O	L18	95	376
I/O	L17	96	379
I/O	L16	97	382
GND	–	98	–
I/O	–	99	–
I/O	M15	100	388
VCC	–	101	–
I/O	M18	102	–
I/O	M17	103	394
I/O	N18	104	397
I/O	P18	105	400
GND	M16	106	–
I/O	N15	107	–
I/O	P15	108	406
I/O	N17	109	409
I/O	R18	110	412
I/O	T18	111	415
I/O	P17	112	418
I/O	N16	113	421
I/O	T17	114	424
I/O	R17	115	427
I/O	P16	116	430
I/O	U18	117	433
SGCK3 (I/O)	T16	118	436
GND	R16	119	–
DONE	U17	120	–
VCC	R15	121	–
$\overline{PROG}$	V18	122	–
I/O (D7)	T15	123	439
PGCK3 (I/O)	U16	124	442
I/O	T14	125	445
I/O	U15	126	448
I/O	R14	127	451
I/O	R13	128	454
I/O (D6)	V17	129	457
I/O	V16	130	460
I/O	T13	131	463
I/O	U14	132	466
I/O	V15	133	469
I/O	V14	134	472
GND	T12	135	–
I/O	R12	136	–
I/O	R11	137	478
I/O	U13	138	481
I/O	V13	139	484
VCC	–	140	–
I/O (D5)	U12	141	487
I/O ($\overline{CSO}$)	V12	142	490
GND	–	143	–
I/O	T11	144	493
I/O	U11	145	496
I/O	V11	146	499
I/O	V10	147	502
I/O (D4)	U10	148	505
I/O	T10	149	508
VCC	R10	150	–
GND	R9	151	–

Pin Description	PG223	MQ240	Bound Scan
I/O (D3	T9	152	511
I/O ($\overline{RS}$)	U9	153	514
I/O	V9	154	517
I/O	V8	155	520
I/O	U8	156	523
I/O	T8	157	526
GND		158	
I/O (D2)	V7	159	529
I/O	U7	160	532
VCC	–	161	–
I/O	V6	162	–
I/O	U6	163	538
I/O	R8	164	541
I/O	R7	165	544
GND	T7	166	–
I/O	R6	167	–
I/O	R5	168	550
I/O	V5	169	553
I/O	V4	170	556
I/O	U5	171	559
I/O	T6	172	562
I/O (D1)	V3	173	565
I/O (RCLK-$\overline{BUSY}$/RDY)	V2	174	568
I/O	U4	175	571
I/O	T5	176	574
I/O (D0, DIN)	U3	177	577
SGCK4 (DOUT, I/O)	T4	178	580
CCLK	V1	179	–
VCC	R4	180	–
O (TDO)	U2	181	–
GND	R3	182	–
I/O (A0, $\overline{WS}$)	T3	183	2
PGCK4 (I/O, A1)	U1	184	5
I/O	P3	185	8
I/O	R2	186	11
I/O (CS1, A2)	T2	187	14
I/O (A3)	N3	188	17
I/O	P4	189	20
I/O	N4	190	23
I/O	P2	191	26
I/O	T1	192	29
I/O	R1	193	32
I/O	N2	194	35
–	–	195*	–
GND	M3	196	–
I/O	P1	197	38
I/O	N1	198	41
I/O	M4	199	44
I/O	L4	200	47
VCC	–	201	–
I/O (A4)	M2	202	50
I/O (A5)	M1	203	53
GND	–	204	–
I/O	–	205	–
I/O	L2	206	59
I/O	L1	207	62
I/O	K1	208	65
I/O (A6)	K2	209	68
I/O (A7)	K3	210	71
GND	K4	211	–

* Indicates unconnected package pins.
† Contributes only one bit (.i) to the boundary scan register.
Boundary Scan Bit 0 = TDO.T
Boundary Scan Bit 1 = TDO.O
Boundary Scan Bit 583 = BSCANT.UPD

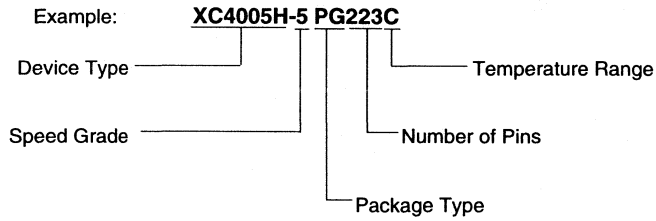
For a detailed description of the device architecture, see page 2-9 through 2-31.

For a detailed description of the configuration modes and their timing, see pages 2-32 through 2-55.

For detailed lists of package pinouts, see pages 2-100 through 2-101.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS		84	100			120	144	156	160	164	191	196	208		223	240	
TYPE		PLAST. PLCC	PLAST. PQFP	PLAST. VQFP	TOP BRAZED CQFP	CERAM. PGA	PLAST. TQFP	CERAM PGA	PLAST. PQFP	TOP BRAZED CQFP	CERAM. PGA	TOP BRAZED CQFP	PLAST. PQFP	METAL PQFP	CERAM. PGA	PLAST. PQFP	METAL PQFP
CODE		PC84	PQ100	VQ100	CB100	PG120	TQ144	PG156	PQ160	CB164	PG191	CB196	PQ208	MQ208	PG223	PQ240	MQ240
XC4003H	-6										C I		C I				
	-5										C		C				
	-4										C		C				
XC4005H	-6														C I	(C I)	C I
	-5														C	(C)	C
	-4														C	(C)	C

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C

B = MIL-STD-883C Class B Parentheses indicates future product plans

XC3000 Logic Cell Array Families

Table of Contents

Overview

Introduced in 1987/88, XC3000 is the industry's most successful family of FPGAs, with over 10 million devices shipped. In 1992/93, Xilinx introduced three additional families, offering more speed, functionality, and a new supply-voltage option.

There are now four distinct family groupings within the XC3000 class of LCA devices.

- XC3000 Family
- XC3000A Family
- XC3000L Family
- XC3100 Family

All four families share a common architecture, development software, design and programming methodology, and also common package pin-outs. An extensive Product Description covers these common aspects. (Page 2-99).

The much shorter individual Product Specifications then provide detailed parametric information for the four individual product families.

Here is a simple overview.

XC3000 Family
The basic XC3000 family forms the cornerstone for the rest of the XC3000 class of devices. The basic XC3000 family offers five different device densities with guaranteed toggle rates from 70 to 125 MHz.

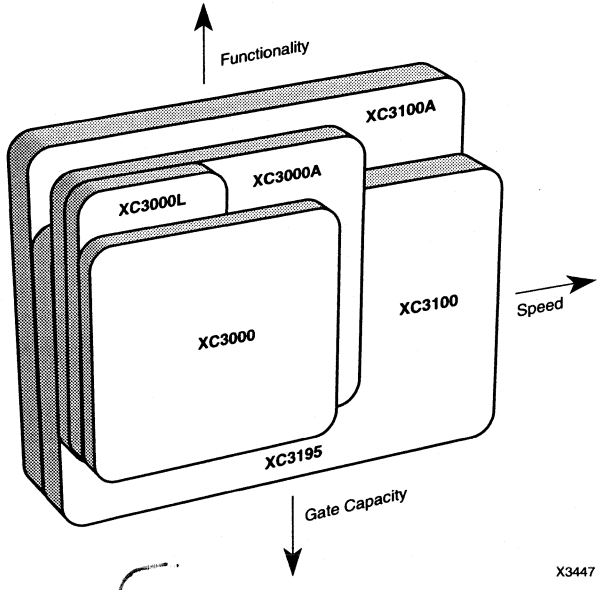
XC3000A Family
The XC3000A is an enhanced version of the basic XC3000 family, featuring additional interconnect resources and other user-friendly enhancements. The ease-of-use of the XC3000A family makes it the obvious choice for all new designs that do not require the speed of the XC3100 or the 3-V operation of the XC3000L.

XC3000L Family
The XC3000L is identical in architecture and features to the XC3000A family, but operates at a nominal supply voltage of 3.3 V. The XC3000L is the right solution for battery-operated and low-power applications.

XC3100 Family
The XC3100 is a performance-optimized relative of the basic XC3000 family. While both families are bitstream and footprint compatible, the XC3100 family extends toggle rates to 270 MHz and in-system performance to 80 MHz. The XC3100 family also offers one additional array size, the XC3195. The XC3100 is best suited for designs that require the highest clock speed or the shortest net delays.

XC3100A Family
The XC3100A combines the enhanced feature set of the XC3000A with the performance of the XC3100. It offers the highest functionality, speed and capacity of all XC3000 families.

The figure below illustrates the relationships between the families. Compared to the original XC3000 family, XC3000A offers additional functionality and, coming soon, increased speed. The XC3000L family offers the same additional functionality, but reduced speed due to its lower supply voltage of 3.3 V. The XC3100 family offers no additional functionality, but substantially higher speed, and higher density with its new member, the XC3195.



XC3000, XC3000A, XC3000L, XC3100, XC3100A Logic Cell Array Families

Product Description

Features

- Complete line of five related Field Programmable Gate Array product families
  - XC3000, XC3000A, XC3000L, XC3100, XC3100A
- Ideal for a wide range of custom VLSI design tasks
  - Replaces TTL, MSI, and other PLD logic
  - Integrates complete sub-systems into a single package
  - Avoids the NRE, time delay, and risk of conventional masked gate arrays
- High-performance CMOS static memory technology
  - Guaranteed toggle rates of 70 to 270 MHz, logic delays from 9 to 3 ns
  - System clock speeds of up to 80 MHz
  - Low quiescent and active power consumption
- Flexible FPGA architecture
  - Compatible arrays ranging from 1,300 to 9,000 gate complexity
  - Extensive register, combinatorial, and I/O capabilities
  - High fan-out signal distribution, low-skew clock nets
  - Internal 3-state bus capabilities
  - TTL or CMOS input thresholds
  - On-chip crystal oscillator amplifier
- Unlimited reprogrammability
  - Easy design iteration
  - In-system logic changes
- Extensive Packaging Options
  - Over 20 different packages
  - Plastic and ceramic surface-mount and pin-grid-array packages
  - Thin and Very Thin Quad Flat Pack (TQFP and VQFP) options
- Ready for volume production
  - Standard, off-the-shelf product availability
  - 100% factory pre-tested devices
  - Excellent reliability record
- Complete XACT Development System
  - Schematic capture, automatic place and route
  - Logic and timing simulation
  - Interactive design editor for design optimization
  - Timing calculator
  - Interfaces to popular design environments like Viewlogic, Cadence, Mentor Graphics, and others

Description

The CMOS XC3000 Class of Logic Cell Array (LCA) families provide a group of high-performance, high-density, digital integrated circuits. Their regular, extendable, flexible, user-programmable array architecture is composed of a configuration program store plus three types of configurable elements: a perimeter of I/O Blocks (IOBs), a core array of Configurable Logic Bocks (CLBs) and resources for interconnection. The general structure of an LCA device is shown in Figure 1 on the next page. The XACT development system provides schematic capture and auto place-and-route for design entry. Logic and timing simulation, and in-circuit emulation are available as design verification alternatives. The design editor is used for interactive design optimization, and to compile the data pattern that represents the configuration program.

The LCA user logic functions and interconnections are determined by the configuration program data stored in internal static memory cells. The program can be loaded in any of several modes to accommodate various system requirements. The program data resides externally in an EEPROM, EPROM or ROM on the application circuit board, or on a floppy disk or hard disk. On-chip initialization logic provides for optional automatic loading of program data at power-up. The companion XC17XX Serial Configuration PROMs provide a very simple serial configuration program storage in a one-time programmable package.

Device	CLBs	Array	User I/Os Max	Flip-Flops	Horizontal Longlines	Configuration Data Bits
XC3020, 3020A, 3020L, 3120, 3120A	64	8 x 8	64	256	16	14,779
XC3030, 3030A, 3030L, 3130, 3130A	100	10 x 10	80	360	20	22,176
XC3042, 3042A, 3042L, 3142, 3142A	144	12 x 12	96	480	24	30,784
XC3064, 3064A, 3064L, 3164, 3164A	224	16 x 14	120	688	32	46,064
XC3090, 3090A, 3090L, 3190, 3190A	320	16 x 20	144	928	40	64,160
XC3195, 3195A	484	22 x 22	176	1,320	44	94,984

The XC3000 Logic Cell Array families provide a variety of logic capacities, package styles, temperature ranges and speed grades.

Architecture

The perimeter of configurable IOBs provides a programmable interface between the internal logic array and the device package pins. The array of CLBs performs user-specified logic functions. The interconnect resources are programmed to form networks, carrying logic signals among blocks, analogous to printed circuit board traces connecting MSI/SSI packages.

The block logic functions are implemented by programmed look-up tables. Functional options are implemented by program-controlled multiplexers. Interconnecting networks between blocks are implemented with metal segments joined by program-controlled pass transistors.

These LCA functions are established by a configuration program which is loaded into an internal, distributed array of configuration memory cells. The configuration program is loaded into the LCA device at power-up and may be reloaded on command. The Logic Cell Array includes logic and control signals to implement automatic or passive configuration. Program data may be either bit serial or byte parallel. The XACT development system generates the configuration program bitstream used to configure the LCA device. The memory loading process is independent of the user logic functions.

Configuration Memory

The static memory cell used for the configuration memory in the Logic Cell Array has been designed specifically for high reliability and noise immunity. Integrity of the LCA device configuration memory based on this design is assured even under adverse conditions. Compared with other programming alternatives, static memory provides the best combination of high density, high performance, high reliability and comprehensive testability. As shown in Figure 2, the basic memory cell consists of two CMOS inverters plus a pass transistor used for writing and reading cell data. The cell is only written during configuration and only read during readback. During normal operation, the cell provides continuous control and the pass transistor is off and does not affect cell stability. This is quite different from the operation of conventional memory devices, in which the cells are frequently read and rewritten.

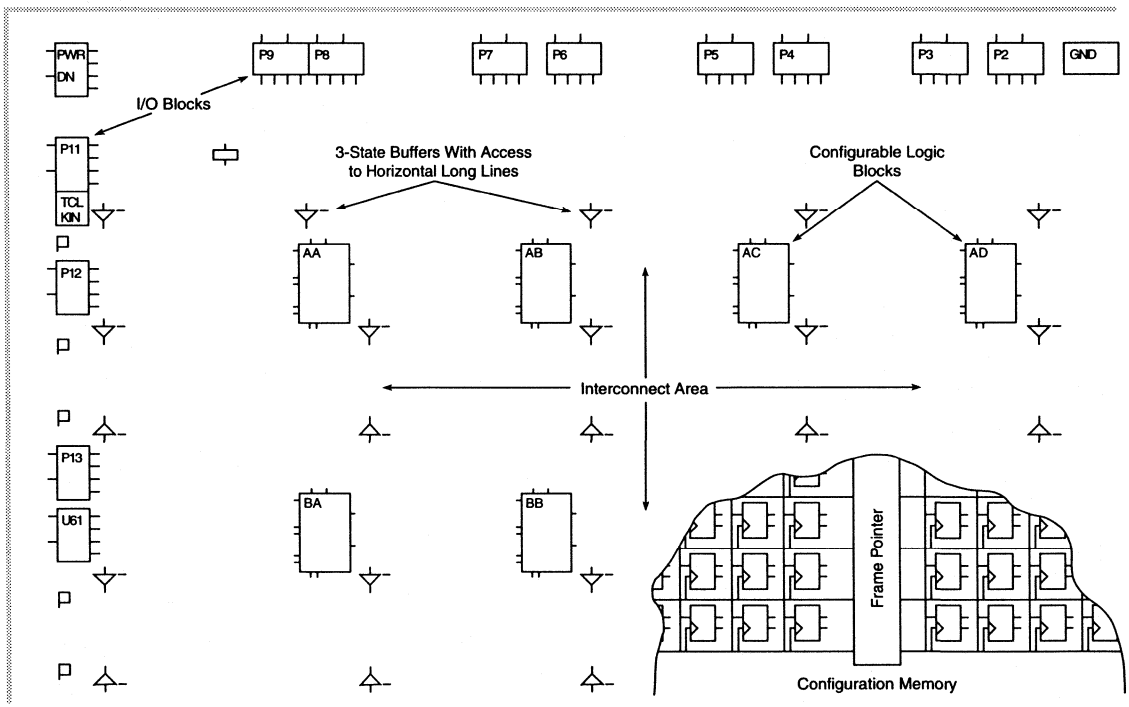


Figure 1. Logic Cell Array Structure. X3241

It consists of a perimeter of programmable I/O blocks, a core of configurable logic blocks and their interconnect resources. These are all controlled by the distributed array of configuration program memory cells.

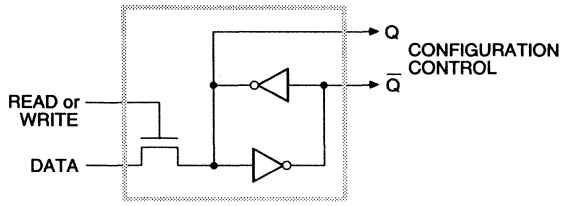


Figure 2. Static Configuration Memory Cell.
It is loaded with one bit of configuration program and controls one program selection in the Logic Cell Array.

The memory cell outputs Q and $\overline{Q}$ use ground and V_{CC} levels and provide continuous, direct control. The additional capacitive load together with the absence of address decoding and sense amplifiers provide high stability to the cell. Due to the structure of the configuration memory cells, they are not affected by extreme power-supply excursions or very high levels of alpha particle radiation. In reliability testing, no soft errors have been observed even in the presence of very high doses of alpha radiation.

The method of loading the configuration data is selectable. Two methods use serial data, while three use byte-wide data. The internal configuration logic utilizes framing information, embedded in the program data by the XACT development system, to direct memory-cell loading. The serial-data framing and length-count preamble provide programming compatibility for mixes of various LCA device devices in a synchronous, serial, daisy-chain fashion.

I/O Block

Each user-configurable IOB shown in Figure 3, provides an interface between the external package pin of the device and the internal user logic. Each IOB includes both registered and direct input paths. Each IOB provides a programmable 3-state output buffer, which may be driven by a registered or direct output signal. Configuration options allow each IOB an inversion, a controlled slew rate and a high impedance pull-up. Each input circuit also provides input clamping diodes to provide electrostatic protection, and circuits to inhibit latch-up produced by input currents.

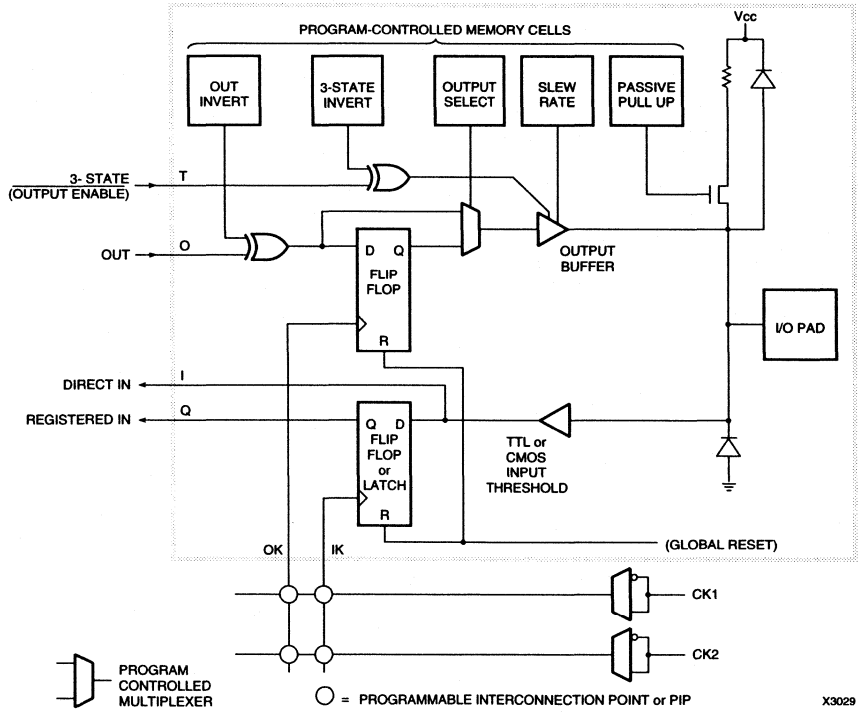


Figure 3. Input/Output Block.
Each IOB includes input and output storage elements and I/O options selected by configuration memory cells. A choice of two clocks is available on each die edge. The polarity of each clock line (not each flip-flop or latch) is programmable. A clock line that triggers the flip-flop on the rising edge is an active Low Latch Enable (Latch transparent) signal and vice versa. Passive pull-up can only be enabled on inputs, not on outputs. All user inputs are programmed for TTL or CMOS thresholds.

The input-buffer portion of each IOB provides threshold detection to translate external signals applied to the package pin to internal logic levels. The global input-buffer threshold of the IOBs can be programmed to be compatible with either TTL or CMOS levels. The buffered input signal drives the data input of a storage element, which may be configured as either a flip-flop or a latch. The clocking polarity (rising/falling edge-triggered flip-flop, High/Low transparent latch) is programmable for each of the two clock lines on each of the four die edges. Note that a clock line driving a *rising* edge-triggered flip-flop makes any latch driven by the same line on the same edge *Low*-level transparent and vice versa (*falling* edge, *High* transparent). All Xilinx primitives in the supported schematic-entry packages, however, are positive edge-triggered flip-flops or High transparent latches. When one clock line must drive flip-flops as well as latches, it is necessary to compensate for the difference in clocking polarities with an additional inverter either in the flip-flop clock input or the latch-enable input. I/O storage elements are reset during configuration or by the active-Low chip $\overline{\text{RESET}}$ input. Both direct input (from IOB pin I) and registered input (from IOB pin Q) signals are available for interconnect.

For reliable operation, inputs should have transition times of less than 100 ns and should not be left floating. Floating CMOS input-pin circuits might be at threshold and produce oscillations. This can produce additional power dissipation and system noise. A typical hysteresis of about 300 mV reduces sensitivity to input noise. Each user IOB includes a programmable high-impedance pull-up resistor, which may be selected by the program to provide a constant High for otherwise undriven package pins. Although the Logic Cell Array provides circuitry to provide input protection for electrostatic discharge, normal CMOS handling precautions should be observed.

Flip-flop loop delays for the IOB and logic-block flip-flops are about 3 ns. This short delay provides good performance under asynchronous clock and data conditions. Short loop delays minimize the probability of a metastable condition that can result from assertion of the clock during data transitions. Because of the short-loop-delay characteristic in the Logic Cell Array, the IOB flip-flops can be used to synchronize external signals applied to the device. Once synchronized in the IOB, the signals can be used internally without further consideration of their clock relative timing, except as it applies to the internal logic and routing-path delays.

IOB output buffers provide CMOS-compatible 4-mA source-or-sink drive for high fan-out CMOS or TTL-compatible signal levels (8 mA in the XC3100 family). The network driving IOB pin O becomes the registered or direct data source for the output buffer. The 3-state control signal (IOB) pin FT can control output activity. An open-drain output may be obtained by using the same signal for driving the output and 3-state signal nets so that the buffer output is enabled only for a Low.

Configuration program bits for each IOB control features such as optional output register, logic signal inversion, and 3-state and slew-rate control of the output.

The program-controlled memory cells of Figure 3 control the following options.

- Logic **inversion of the output** is controlled by one configuration program bit per IOB.

- Logic **3-state control** of each IOB output buffer is determined by the states of configuration program bits which turn the buffer on, or off, or select the output buffer 3-state control interconnection (IOB pin T). When this IOB output control signal is High, a logic one, the buffer is **disabled** and the package pin is high impedance. When this IOB output control signal is Low, a logic zero, the buffer is **enabled** and the package pin is active. Inversion of the buffer 3-state control-logic sense (output enable) is controlled by an additional configuration program bit.

- **Direct or registered output** is selectable for each IOB. The register uses a positive-edge, clocked flip-flop. The clock source may be supplied (IOB pin OK) by either of two metal lines available along each die edge. Each of these lines is driven by an invertible buffer.

- Increased **output transition speed** can be selected to improve critical timing. Slower transitions reduce capacitive-load peak currents of non-critical outputs and minimize system noise.

- An internal high-impedance **pull-up resistor** (active by default) prevents unconnected inputs from floating.

Summary of I/O Options

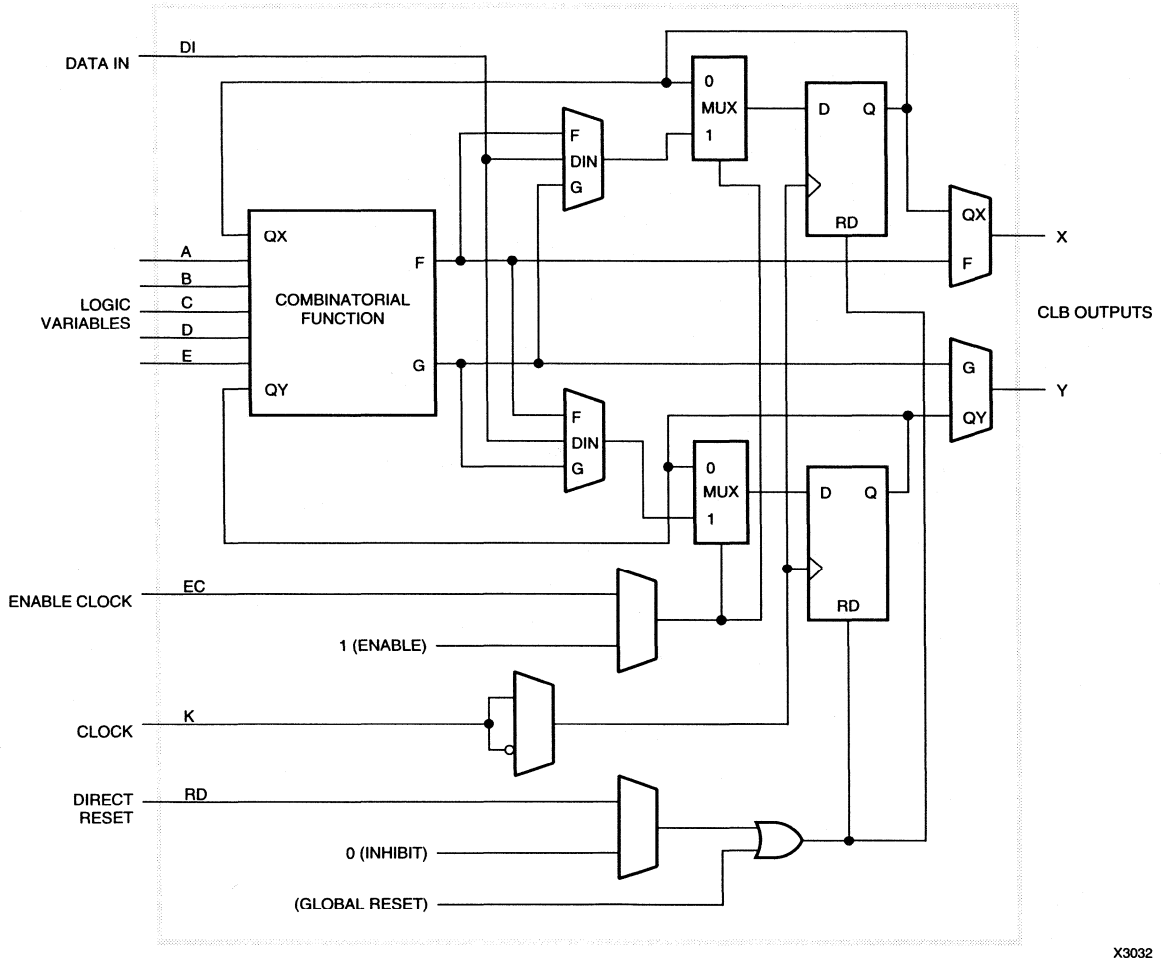
- Inputs
  - Direct
  - Flip-flop/latch
  - CMOS/TTL threshold (chip inputs)
  - Pull-up resistor/open circuit
- Outputs
  - Direct/registered
  - Inverted/not
  - 3-state/on/off
  - Full speed/slew limited
  - 3-state/output enable (inverse)

Configurable Logic Block

The array of CLBs provides the functional elements from which the user's logic is constructed. The logic blocks are arranged in a matrix within the perimeter of IOBs. The XC3020 has 64 such blocks arranged in 8 rows and 8 columns. The XACT development system is used to compile the configuration data which is to be loaded into the internal configuration memory to define the operation and interconnection of each block. User definition of CLBs and their interconnecting networks may be done by automatic translation from a schematic-capture logic diagram or optionally by installing library or user macros.

Each CLB has a combinatorial logic section, two flip-flops, and an internal control section. See Figure 4. There are: five logic inputs (A, B, C, D and E); a common clock input (K); an asynchronous direct RESET input (RD); and an enable clock (EC). All may be driven from the interconnect resources adjacent to the blocks. Each CLB also has two outputs (X and Y) which may drive interconnect networks.

Data input for either flip-flop within a CLB is supplied from the function F or G outputs of the combinatorial logic, or the block input, DI. Both flip-flops in each CLB share the



X3032

Figure 4. Configurable Logic Block. Each CLB includes a combinatorial logic section, two flip-flops and a program memory controlled multiplexer selection of function. It has.
five logic variable inputs A, B, C, D, and E
a direct data in DI
an enable clock EC
a clock (invertible) K
an asynchronous direct RESET RD
two outputs X and Y

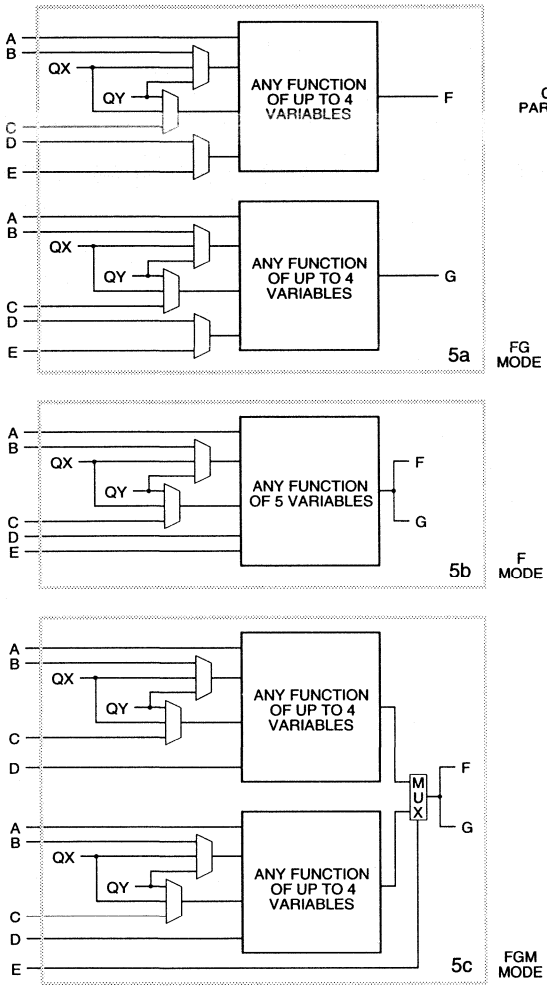


Figure 5

5a. Combinatorial Logic Option FG generates two functions of four variables each. One variable, A, must be common to both functions. The second and third variable can be any choice of of B, C, QX and QY The fourth variable can be any choice of D or E.

5b. Combinatorial Logic Option F generates any function of five variables: A, D, E and and two choices out of B, C, QX, QY.

5c. Combinatorial Logic Option FGM allows variable E to select between two functions of four variables: Both have common inputs A and D and any choice out of B, C, QX and QY for the remaining two variables. Option 3 can then implement some functions of six or seven variables.

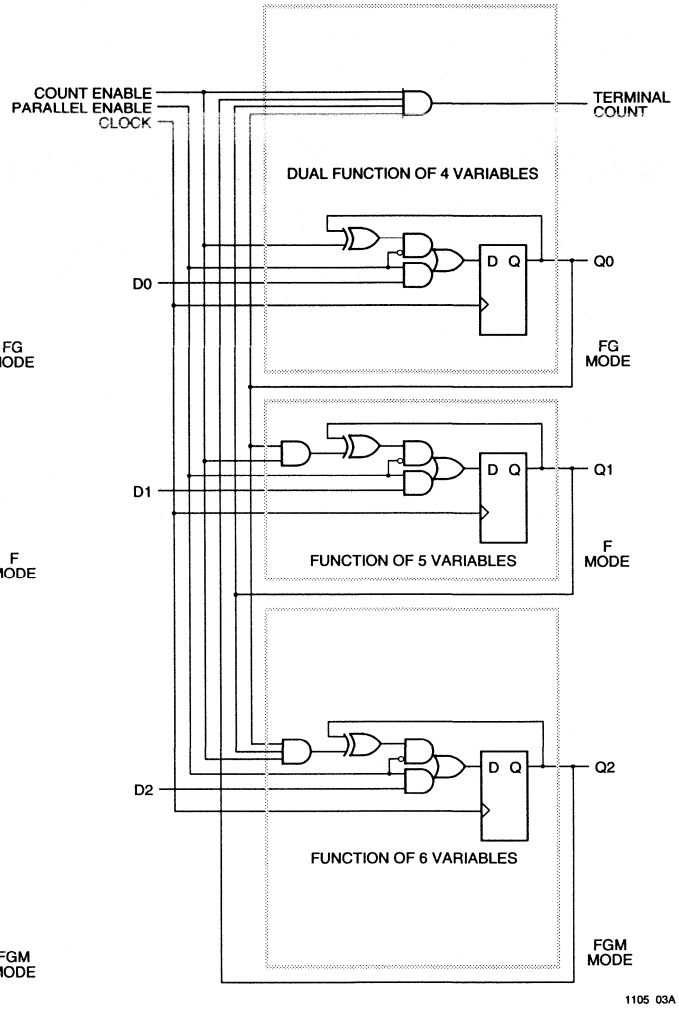


Figure 6. C8BCP Macro.

The C8BCP macro (modulo-8 binary counter with parallel enable and clock enable) uses one combinatorial logic block of each option.

asynchronous RD which, when enabled and High, is dominant over clocked inputs. All flip-flops are reset by the active-Low chip input, $\overline{\text{RESET}}$, or during the configuration process. The flip-flops share the enable clock (EC) which, when Low, recirculates the flip-flops' present states and inhibits response to the data-in or combinatorial function inputs on a CLB. The user may enable these control inputs and select their sources. The user may also select the clock net input (K), as well as its active sense within each CLB. This programmable inversion eliminates the need to route both phases of a clock signal throughout the device. Flexible routing allows use of common or individual CLB clocking.

The combinatorial-logic portion of the CLB uses a 32 by 1 look-up table to implement Boolean functions. Variables selected from the five logic inputs and two internal block flip-flops are used as table address inputs. The combinatorial propagation delay through the network is independent of the logic function generated and is spike free for single input variable changes. This technique can generate two independent logic functions of up to four variables each as shown in Figure 5a, or a single function of five variables as shown in Figure 5b, or some functions of seven variables as shown in Figure 5c. Figure 6 shows a modulo-8 binary counter with parallel enable. It uses one CLB of each type. The partial functions of six or seven variables are implemented using the input variable (E) to dynamically select between two functions of four different variables. For the two functions of four variables each, the independent results (F and G) may be used as data inputs to either flip-flop or either logic block output. For the single function of five variables and merged functions of six or seven variables, the F and G outputs are identical. Symmetry of the F and G functions and the flip-flops allows the interchange of CLB outputs to optimize routing efficiencies of the networks interconnecting the CLBs and IOBs.

Programmable Interconnect

Programmable-interconnection resources in the Logic Cell Array provide routing paths to connect inputs and outputs of the IOBs and CLBs into logic networks. Interconnections between blocks are composed of a two-layer grid of metal segments. Specially designed pass transistors, each controlled by a configuration bit, form programmable interconnect points (PIPs) and switching matrices used to implement the necessary connections between selected metal segments and block pins. Figure 7 is an example of a routed net. The XACT development system provides automatic routing of these interconnections. Interactive routing (Editnet) is also available for design optimization. The inputs of the CLBs or IOBs are multiplexers which can be programmed to select an input network from the adjacent interconnect segments. ***Since the switch connections to block inputs are unidirectional, as are block outputs, they are usable only for block input connection and not for routing.*** Figure 8 illustrates routing access to logic block input variables, control inputs and block outputs. Three types of metal resources are provided to accommodate various network interconnect requirements.

- General Purpose Interconnect
- Direct Connection
- Longlines (multiplexed busses and wide AND gates)

General Purpose Interconnect

General purpose interconnect, as shown in Figure 9, consists of a grid of five horizontal and five vertical metal segments located between the rows and columns of logic and IOBs. Each segment is the height or width of a logic block. Switching matrices join the ends of these segments and allow programmed interconnections between the metal grid segments of adjoining rows and columns. The switches of an unprogrammed device are all non-conducting. The connections through the switch matrix may be established by the automatic routing or by using Editnet to select the desired pairs of matrix pins to be connected or disconnected. The legitimate switching matrix combinations for each pin are indicated in Figure 10 and may be highlighted by the use of the Show-Matrix command in the XACT system.

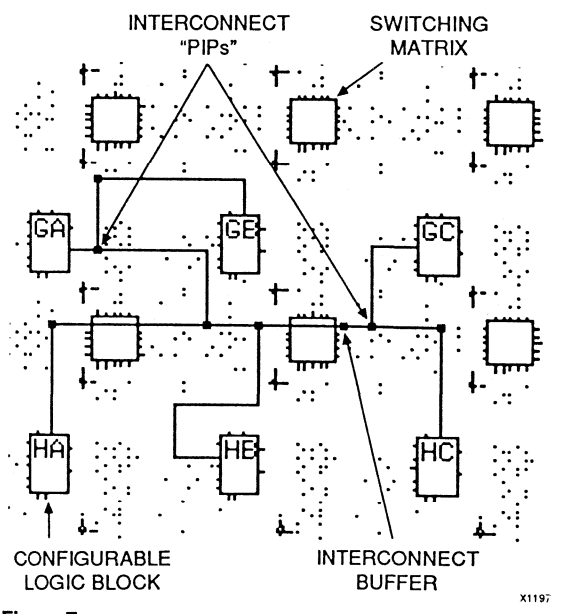


Figure 7.
An XACT view of routing resources used to form a typical interconnection network from CLB GA.

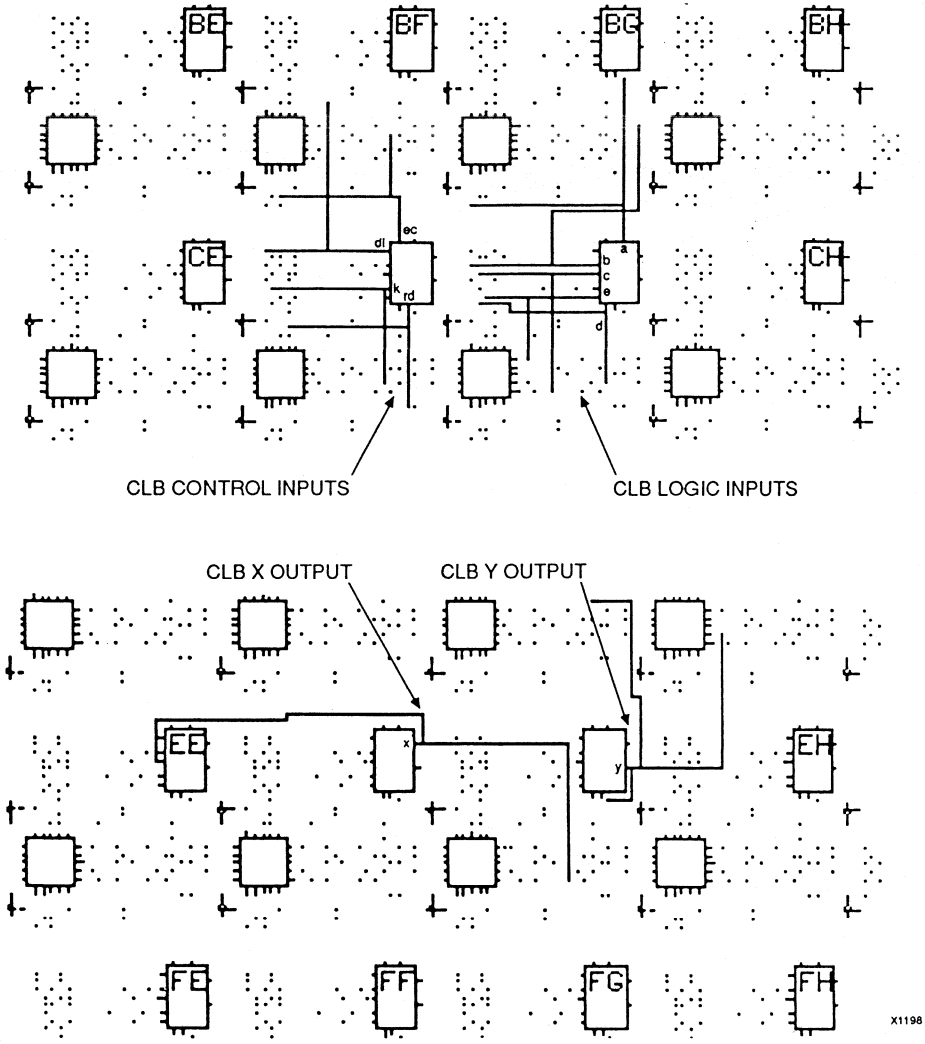


Figure 8. XACT Development System Locations of interconnect access, CLB control inputs, logic inputs and outputs. The dot pattern represents the available programmable interconnection points (PIPs).

Some of the interconnect PIPs are directional. This is indicated on the XACT design editor status line:

ND is a nondirectional interconnection.
D:H->V is a PIP that drives from a horizontal to a vertical line.
D:V->H is a PIP that drives from a vertical to a horizontal line.
D:C->T is a "T" PIP that drives from a cross of a T to the tail.
D:CW is a corner PIP that drives in the clockwise direction.
P0 indicates the PIP is non-conducting , P1 is on.

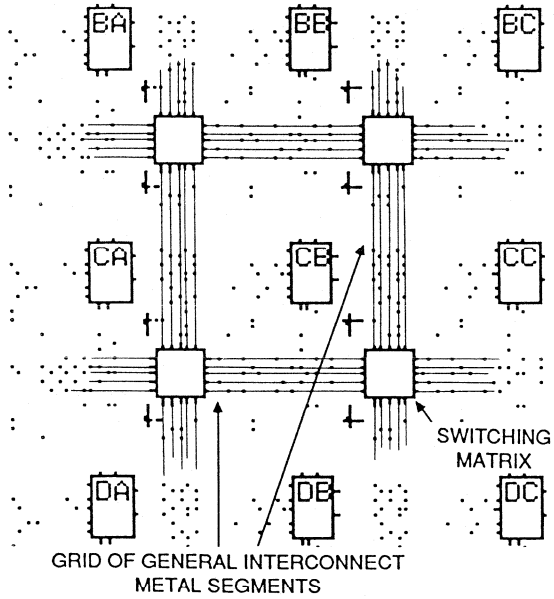


Figure 9. LCA General-Purpose Interconnect.
Composed of a grid of metal segments that may be interconnected through switch matrices to form networks for CLB and IOB inputs and outputs. X2664

Special buffers within the general interconnect areas provide periodic signal isolation and restoration for improved performance of lengthy nets. The interconnect buffers are available to propagate signals in either direction on a given general interconnect segment. These bidirectional (bidi) buffers are found adjacent to the switching matrices, above and to the right and may be highlighted by the use of the Show BIDI command in the XACT system. The other PIPs adjacent to the matrices are accessed to or from Longlines. The development system automatically defines the buffer direction based on the location of the interconnection network source. The delay calculator of the XACT development system automatically calculates and displays the block, interconnect and buffer delays for any paths selected. Generation of the simulation netlist with a worst-case delay model is provided by an XACT option.

Direct Interconnect

Direct interconnect, shown in Figure 11, provides the most efficient implementation of networks between adjacent CLBs or I/O Blocks. Signals routed from block to block using the direct interconnect exhibit minimum interconnect propagation and use no general interconnect resources. For each CLB, the X output may be connected directly to the B input of the CLB immediately to its right and to the C input of the CLB to its left. The Y output can use direct interconnect to drive the D input of the block immediately above and the A input of the block below. Direct intercon-

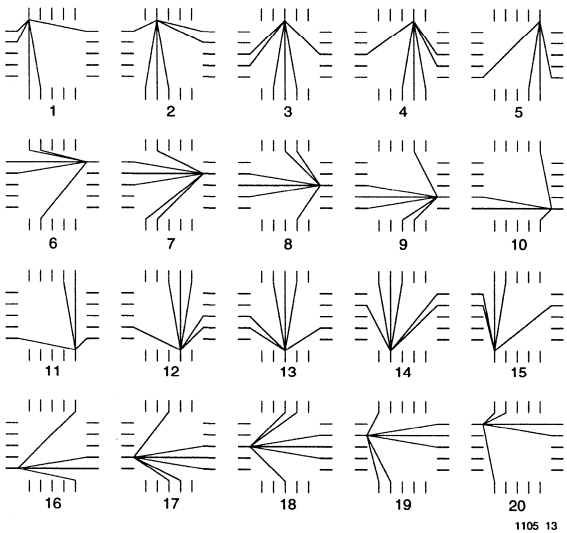


Figure 10. Switch Matrix Interconnection Options for Each Pin. Switch matrices on the edges are different. Use Show Matrix menu option in the XACT system

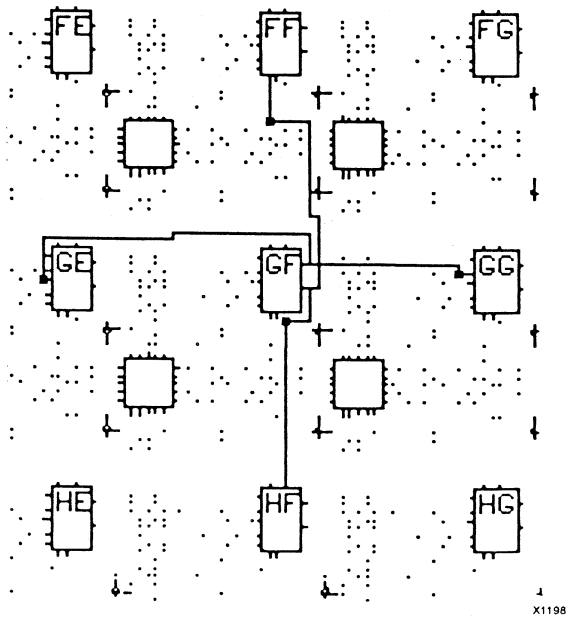


Figure 11. CLB X and Y Outputs.
The X and Y outputs of each CLB have single contact, direct access to inputs of adjacent CLBs

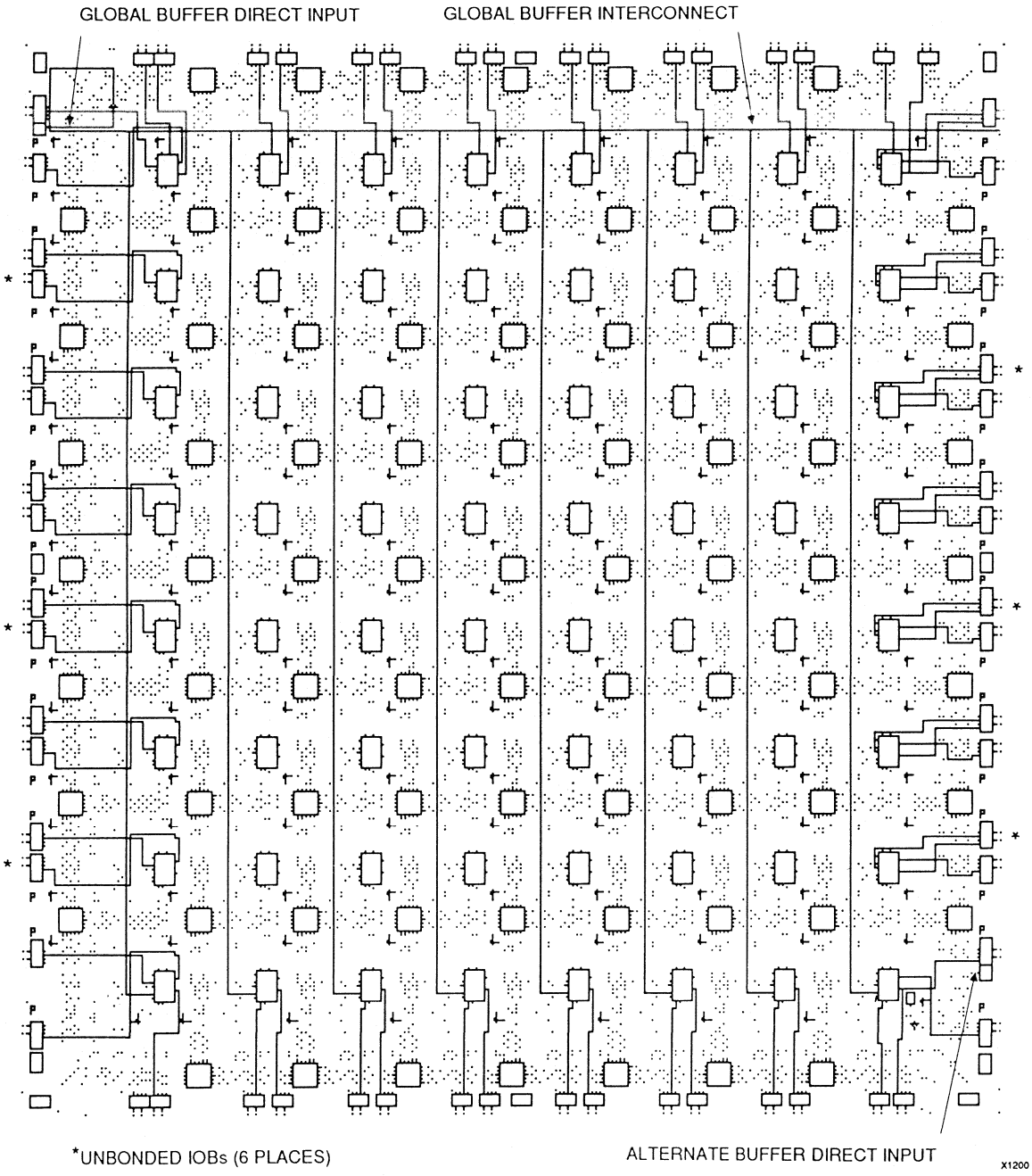


Figure 12. XC3020 Die-Edge IOBs. The XC3020 die-edge IOBs are provided with direct access to adjacent CLBs.

nect should be used to maximize the speed of high-performance portions of logic. Where logic blocks are adjacent to IOBs, direct connect is provided alternately to the IOB inputs (I) and outputs (O) on all four edges of the die. The right edge provides additional direct connects from CLB outputs to adjacent IOBs. Direct interconnections of IOBs with CLBs are shown in Figure 12.

Longlines

The Longlines bypass the switch matrices and are intended primarily for signals that must travel a long distance, or must have minimum skew among multiple destinations. Longlines, shown in Figure 13, run vertically and horizontally the height or width of the interconnect area. Each interconnection column has three vertical Longlines, and each interconnection row has two horizontal Longlines. Two additional Longlines are located adjacent to the outer sets of switching matrices. In devices larger than the XC3020, two vertical Longlines in each column are connectable half-length lines. On the XC3020, only the outer Longlines are connectable half-length lines.

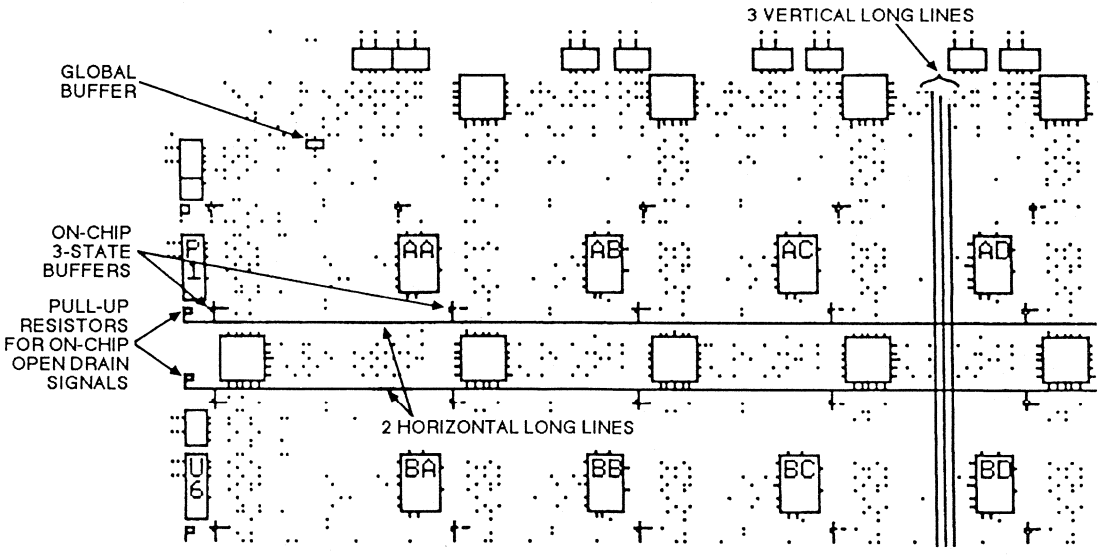
Longlines can be driven by a logic block or IOB output on a column-by-column basis. This capability provides a common low skew control or clock line within each column of logic blocks. Interconnections of these Longlines are shown in Figure 14. Isolation buffers are provided at each input to a Longline and are enabled automatically by the development system when a connection is made.

A buffer in the upper left corner of the LCA chip drives a global net which is available to all K inputs of logic blocks. Using the global buffer for a clock signal provides a skew-free, high fan-out, synchronized clock for use at any or all of the IOBs and CLBs. Configuration bits for the K input to each logic block can select this global line or another routing resource as the clock source for its flip-flops. This net may also be programmed to drive the die edge clock lines for IOB use. An enhanced speed, CMOS threshold, direct access to this buffer is available at the second pad from the top of the left die edge.

A buffer in the lower right corner of the array drives a horizontal Longline that can drive programmed connections to a vertical Longline in each interconnection column. This alternate buffer also has low skew and high fan-out. The network formed by this alternate buffer's Longlines can be selected to drive the K inputs of the CLBs. CMOS threshold, high speed access to this buffer is available from the third pad from the bottom of the right die edge.

Internal Busses

A pair of 3-state buffers, located adjacent to each CLB, permits logic to drive the horizontal Longlines. Logic operation of the 3-state buffer controls allows them to implement wide multiplexing functions. Any 3-state buffer input can be selected as drive for the horizontal long-line bus by applying a Low logic level on its 3-state control line. See Figure 15a. The user is required to avoid contention which can result from multiple drivers with opposing logic



X1243

Figure 13. Horizontal and Vertical Longlines. These Longlines provide high fan-out, low-skew signal distribution in each row and column. The global buffer in the upper left die corner drives a common line throughout the LCA device.

levels. Control of the 3-state input by the same signal that drives the buffer input, creates an open-drain wired-AND function. A logic High on both buffer inputs creates a high impedance, which represents no contention. A logic Low enables the buffer to drive the Longline Low. See Figure 15b. Pull-up resistors are available at each end of the Longline to provide a High output when all connected buffers are non-conducting. This forms fast, wide gating functions. When data drives the inputs, and separate signals drive the 3-state control lines, these buffers form multiplexers (3-state busses). In this case, care must be used to prevent contention through multiple active buffers

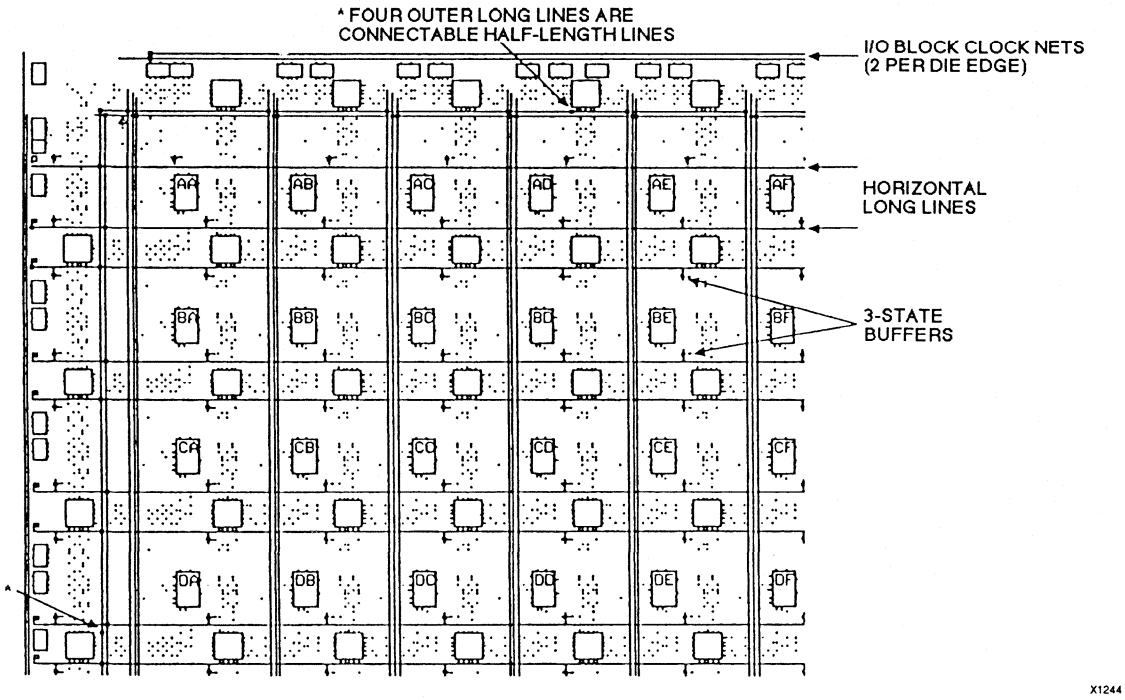


Figure 14. Programmable Interconnection of Longlines. This is provided at the edges of the routing area. Three-state buffers allow the use of horizontal Longlines to form on-chip wired AND and multiplexed buses. The left two non-clock vertical Longlines per column (except XC3020) and the outer perimeter Longlines may be programmed as connectible half-length lines.

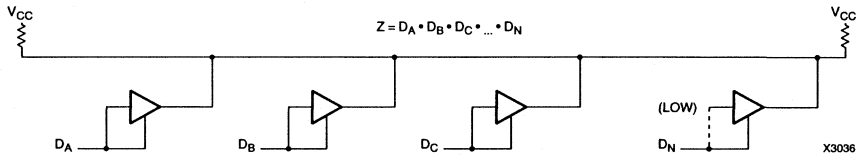
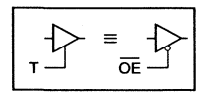


Figure 15a. 3-State Buffers Implement a Wired-AND Function. When all the buffer 3-state lines are High, (high impedance), the pull-up resistor(s) provide the High output. The buffer inputs are driven by the control signals or a Low.



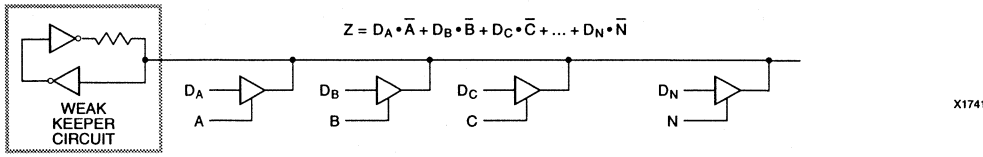


Figure 15b. 3-State Buffers Implement a Multiplexer. The selection is accomplished by the buffer 3-state signal.

of conflicting levels on a common line. Each horizontal Longline is also driven by a weak keeper circuit that prevents undefined floating levels by maintaining the previous logic level when the line is not driven by an active buffer or a pull-up resistor. Figure 16 shows 3-state buffers, Longlines and pull-up resistors.

Crystal Oscillator

Figure 16 also shows the location of an internal high speed inverting amplifier which may be used to implement an on-chip crystal oscillator. It is associated with the auxiliary buffer in the lower right corner of the die. When the oscillator is configured by MakeBits and connected as a signal source, two special user IOBs are also configured to connect the oscillator amplifier with external crystal oscillator components as shown in Figure 17. A divide by two option is available to assure symmetry. The oscillator circuit becomes active early in the configuration process to allow the oscillator to stabilize. Actual internal connection is delayed until completion of configuration. In Figure 17 the feedback resistor R1, between the output and input, biases the amplifier at threshold. The inversion of the amplifier, together with the R-C networks and an AT-cut series resonant crystal, produce the 360-degree phase shift of the Pierce oscillator. A series resistor R2 may be included to add to the amplifier output impedance when needed for phase-shift control, crystal resistance matching, or to limit the amplifier input swing to control clipping at large amplitudes. Excess feedback voltage may be corrected by the ratio of C2/C1. The amplifier is designed to be used from 1 MHz to about one-half the specified CLB toggle frequency. Use at frequencies below 1 MHz may require individual characterization with respect to a series

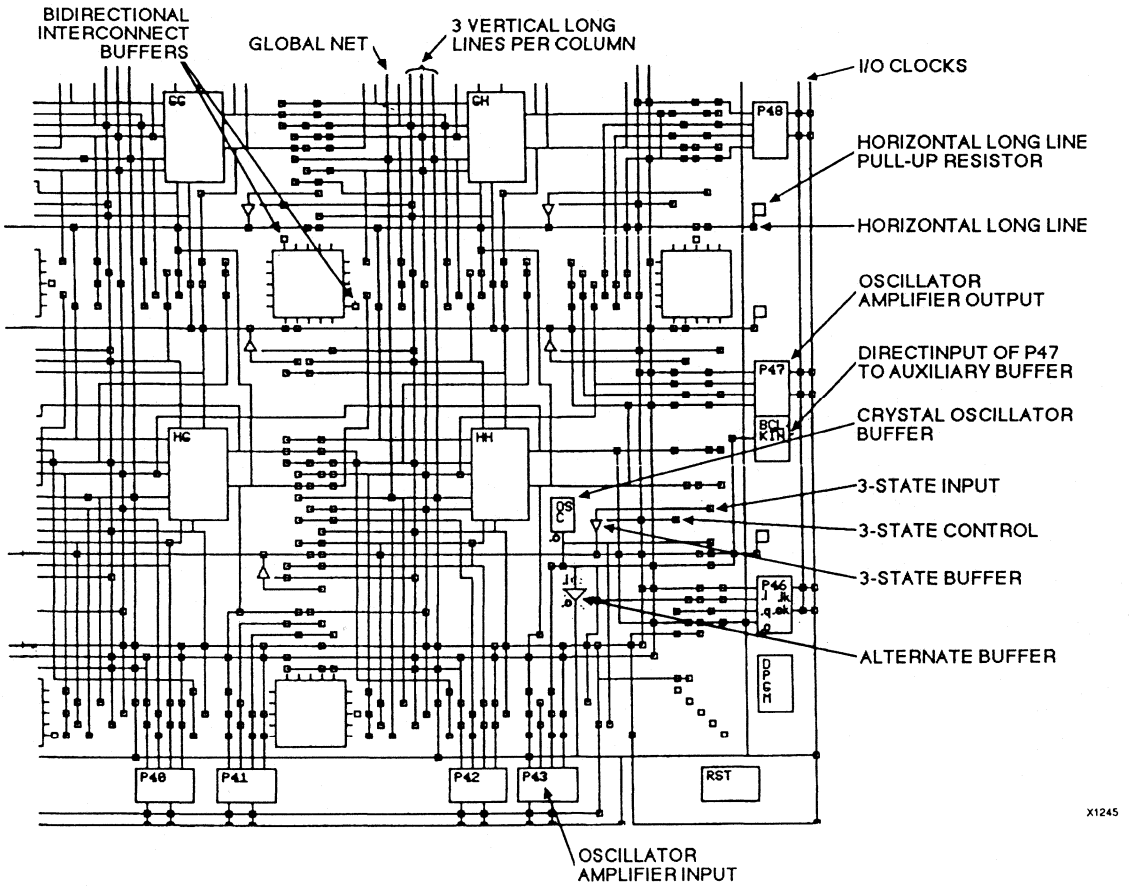
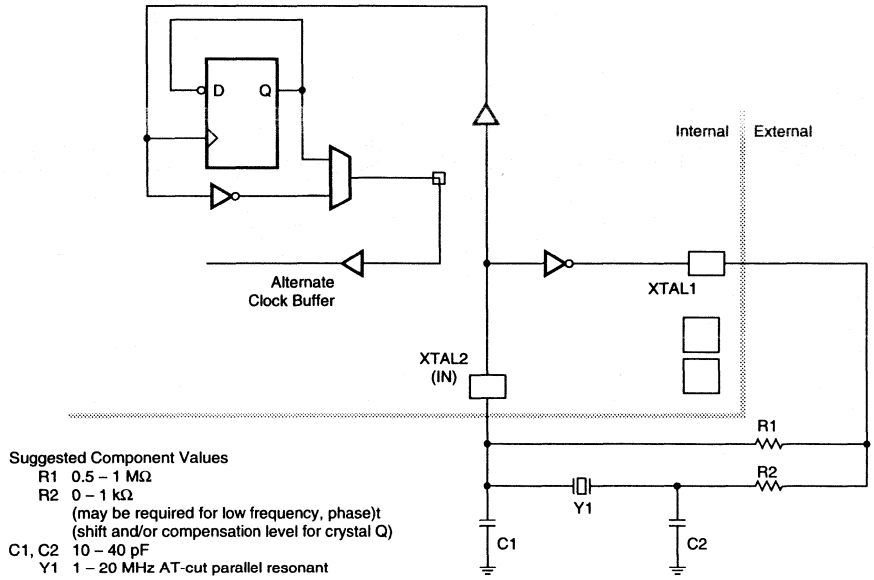


Figure 16. XACT Development System. An extra large view of possible interconnections in the lower right corner of the XC3020.



	44 PIN	68 PIN	84 PIN		100 PIN		132 PIN	160 PIN	164 PIN	175 PIN	208 PIN
	PLCC	PLCC	PLCC	PGA	CQFP	PQFP	PGA	PQFP	CQFP	PGA	PQFP
XTAL 1 (OUT)	30	47	57	J11	67	82	P13	82	105	T14	110
XTAL 2 (IN)	26	43	53	L11	61	76	M13	76	99	P15	100

X5302

Figure 17. Crystal Oscillator Inverter. When activated in the MakeBits program and by selecting an output network for its buffer, the crystal oscillator inverter uses two unconfigured package pins and external components to implement an oscillator. An optional divide-by-two mode is available to assure symmetry.

resistance. Crystal oscillators above 20 MHz generally require a crystal which operates in a third overtone mode, where the fundamental frequency must be suppressed by an inductor across C2, turning this parallel resonant circuit to double the fundamental crystal frequency, i.e., 2/3 of the desired third harmonic frequency network. When the oscillator inverter is not used, these IOBs and their package pins are available for general user I/O.

Programming

Table 1

M0	M1	M2	CCLK	Mode	Data
0	0	0	output	Master	Bit Serial
0	0	1	output	Master	Byte Wide Addr. = 0000 up
0	1	0	—	reserved	—
0	1	1	output	Master	Byte Wide Addr. = FFFF down
1	0	0	—	reserved	—
1	0	1	output	Peripheral	Byte Wide
1	1	0	—	reserved	—
1	1	1	input	Slave	Bit Serial

Initialization Phase

An internal power-on-reset circuit is triggered when power is applied. When Vcc reaches the voltage at which portions of the LCA device begin to operate (nominally 2.5 to 3 V), the programmable I/O output buffers are 3-stated and a high-impedance pull-up resistor is provided for the user I/O pins. A time-out delay is initiated to allow the power supply voltage to stabilize. During this time the power-down mode is inhibited. The Initialization state time-out (about 11 to 33 ms) is determined by a 14-bit counter driven by a self-generated internal timer. This nominal 1-MHz timer is subject to variations with process, temperature and power supply. As shown in Table 1, five configuration mode choices are available as determined by the input levels of three mode pins; M0, M1 and M2.

In Master configuration modes, the LCA device becomes the source of the Configuration Clock (CCLK). The beginning of configuration of devices using Peripheral or Slave modes must be delayed long enough for their initialization to be completed. An LCA device with mode lines selecting a Master configuration mode extends its initialization state using four times the delay (43 to 130 ms) to assure that all daisy-chained slave devices, which it may be driving, will

be ready even if the master is very fast, and the slave(s) very slow. Figure 18 shows the state sequences. At the end of Initialization, the LCA device enters the Clear state where it clears the configuration memory. The active Low, open-drain initialization signal $\overline{INIT}$ indicates when the Initialization and Clear states are complete. The LCA device tests for the absence of an external active Low $\overline{RESET}$ before it makes a final sample of the mode lines and enters the Configuration state. An external wired-AND of one or more $\overline{INIT}$ pins can be used to control configuration by the assertion of the active-Low $\overline{RESET}$ of a master mode device or to signal a processor that the LCA devices are not yet initialized.

If a configuration has begun, a re-assertion of $\overline{RESET}$ for a minimum of three internal timer cycles will be recognized and the LCA device will initiate an abort, returning to the Clear state to clear the partially loaded configuration memory words. The LCA device will then resample $\overline{RESET}$ and the mode lines before re-entering the Configuration state.

A re-program is initiated.when a configured XC3000 family device senses a High-to-Low transition and subsequent >6 µs Low level on the Done/$\overline{PROG}$ package pin, or, if this pin is externally held permanently Low, a High-to-Low transition and subsequent >6 µs Low time on the $\overline{RESET}$ package pin.

The LCA device returns to the Clear state where the configuration memory is cleared and mode lines re-sampled, as for an aborted configuration. The complete configuration program is cleared and loaded during each configuration program cycle.

Length count control allows a system of multiple Logic Cell Arrays, of assorted sizes, to begin operation in a synchronized fashion. The configuration program generated by the MakePROM program of the XACT development system begins with a preamble of 111111110010 followed by a 24-bit length count representing the total number of configuration clocks needed to complete loading of the configuration program(s). The data framing is shown in Figure 19. All LCA devices connected in series read and shift preamble and length count in on positive and out on negative configuration clock edges. An LCA device which has received the preamble and length count then presents a High Data Out until it has intercepted the appropriate number of data frames. When the configuration program memory of an LCA device is full and the length count does not yet compare, the LCA device shifts any additional data through, as it did for preamble and length count.

When the LCA device configuration memory is full and the length count compares, the LCA device will execute a synchronous start-up sequence and become operational. See Figure 20. Two CCLK cycles after the completion of loading configuration data, the user I/O pins are enabled as configured. As selected in MakeBits, the internal user-logic RESET is released either one clock cycle before or after the I/O pins become active. A similar timing selection is programmable for the DONE/$\overline{PROG}$ output signal. DONE/$\overline{PROG}$ may also be programmed to be an open drain or include a pull-up resistor to accommodate wired ANDing. The High During Configuration (HDC) and Low During Configuration ($\overline{LDC}$) are two user I/O pins which are driven active while an LCA device is in its Initialization,

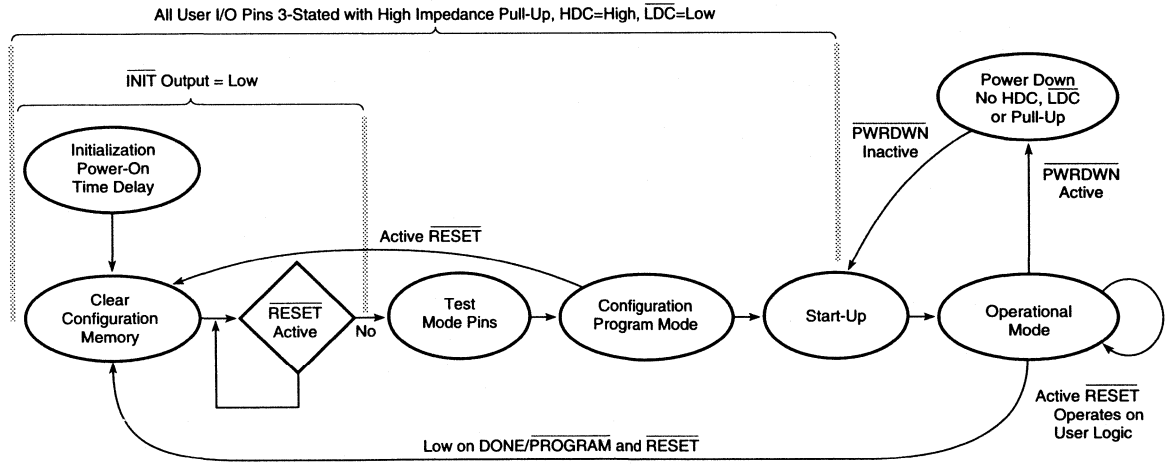


Figure 18. A State Diagram of the Configuration Process for Power-up and Reprogram.

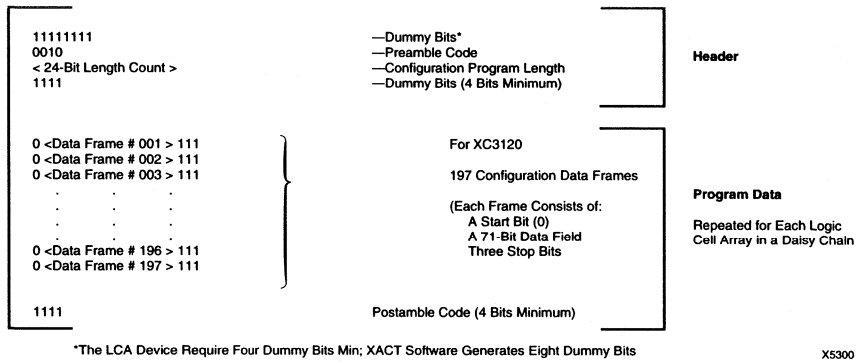
Clear or Configure states. They and DONE/$\overline{PROG}$ provide signals for control of external logic signals such as RESET, bus enable or PROM enable during configuration. For parallel Master configuration modes, these signals provide PROM enable control and allow the data pins to be shared with user logic signals.

User I/O inputs can be programmed to be either TTL or CMOS compatible thresholds. At power-up, all inputs have TTL thresholds and can change to CMOS thresholds at the completion of configuration if the user has selected CMOS thresholds. The threshold of $\overline{PWRDWN}$ and the direct clock inputs are fixed at a CMOS level.

If the crystal oscillator is used, it will begin operation before configuration is complete to allow time for stabilization before it is connected to the internal circuitry.

Configuration Data

Configuration data to define the function and interconnection within a Logic Cell Array is loaded from an external



Device	XC3020 XC3020A XC3020L XC3120 XC3120A	XC3030 XC3030A XC3030L XC3130 XC3130A	XC3042 XC3042A XC3042L XC3142 XC3142A	XC3064 XC3064A XC3064L XC3164 XC3164A	XC3090 XC3090A XC3090L XC3190 XC3190A	XC3195 XC3195A
Gates	1,000 to 1,500	1,500 to 2,000	2,000 to 3,000	3,500 to 4,500	5,000 to 6,000	6,500 to 7,500
CLBs Row x Col	64 (8 x 8)	100 (10 x 10)	144 (12 x 12)	224 (16 x 14)	320 (20 x 16)	484 (22 x 22)
IOBs	64	80	96	120	144	176
Flip-flops	256	360	480	688	928	1,320
Horizontal Longlines	16	20	24	32	40	44
TBUFs/Horizontal LL	9	11	13	15	17	23
Bits per Frame (including1 start and 3 stop bits)	75	92	108	140	172	188
Frames	197	241	285	329	373	505
Program Data = Bits x Frames + 4 bits (excludes header)	14,779	22,176	30,784	46,064	64,160	94,944
PROM size (bits) = Program Data + 40-bit Header	14,819	22,216	30,824	46,104	64,200	94,984

Figure 19. Internal Configuration Data Structure for an LCA Device. This shows the preamble, length count and data frames generated by the XACT Development System.

The Length Count produced by the MakeBits program = [(40-bit preamble + sum of program data + 1 per daisy chain device) rounded up to multiple of 8] – ($2 \le K \le 4$) where K is a function of DONE and RESET timing selected. An additional 8 is added if roundup increment is less than K. K additional clocks are needed to complete start-up after length count is reached.

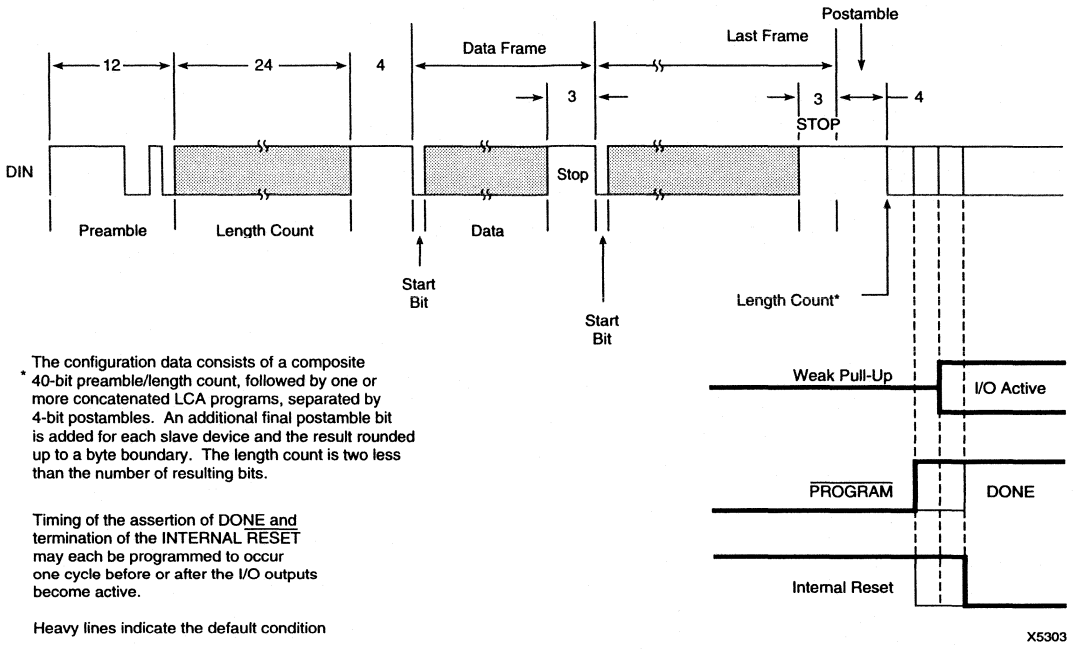


Figure 20. Configuration and Start-up of One or More LCA Devices.

storage at power-up and after a re-program signal. Several methods of automatic and controlled loading of the required data are available. Logic levels applied to mode selection pins at the start of configuration time determine the method to be used. See Table 1. The data may be either bit-serial or byte-parallel, depending on the configuration mode. The different LCA devices have different sizes and numbers of data frames. To maintain compatibility between various device types, the Xilinx product families use compatible configuration formats. For the XC3020, configuration requires 14779 bits for each device, arranged in 197 data frames. An additional 40 bits are used in the header. See Figure 20. The specific data format for each device is produced by the MakeBits command of the development system and one or more of these files can then be combined and appended to a length count preamble and be transformed into a PROM format file by the MakePROM command of the XACT development system. A compatibility exception precludes the use of an XC2000-series device as the master for XC3000-series devices if their DONE or RESET are programmed to occur after their outputs become active.

The Tie Option of the MakeBits program defines output levels of unused blocks of a design and connects these to unused routing resources. This prevents indeterminate levels that might produce parasitic supply currents. If unused blocks are not sufficient to complete the tie, the Flagnet command of EDITLCA can be used to indicate nets which must not be used to drive the remaining unused routing, as that might affect timing of user nets. Norestore will retain the results of tie for timing analysis with Querynet before Restore returns the design to the untied condition. Tie can be omitted for quick breadboard iterations where a few additional milliamps of I_{CC} are acceptable.

The configuration bitstream begins with eight High preamble bits, a 4-bit preamble code and a 24-bit length count. When configuration is initiated, a counter in the LCA device is set to zero and begins to count the total number of configuration clock cycles applied to the device. As each configuration data frame is supplied to the LCA device, it is internally assembled into a data word, which is then loaded in parallel into one word of the internal configuration memory array. The configuration loading process is complete when the current length count equals the loaded length count and the required configuration program data frames have been written. Internal user flip-flops are held Reset during configuration.

Two user-programmable pins are defined in the unconfigured Logic Cell array. High During Configuration (HDC) and Low During Configuration ($\overline{\text{LDC}}$) as well as DONE/$\overline{\text{PROG}}$ may be used as external control signals during configuration. In Master mode configurations it is convenient to use $\overline{\text{LDC}}$ as an active-Low EPROM Chip

Enable. After the last configuration data bit is loaded and the length count compares, the user I/O pins become active. Options in the MakeBits program allow timing choices of one clock earlier or later for the timing of the end of the internal logic RESET and the assertion of the DONE signal. The open-drain DONE/$\overline{PROG}$ output can be AND-tied with multiple LCA devices and used as an active-High READY, an active-Low PROM enable or a RESET to other portions of the system. The state diagram of Figure 18 illustrates the configuration process.

Master Mode

In Master mode, the LCA device automatically loads configuration data from an external memory device. There are three Master modes that use the internal timing source to supply the configuration clock (CCLK) to time the incoming data. Master Serial mode uses serial configuration data supplied to Data-in (DIN) from a synchronous serial source such as the Xilinx Serial Configuration PROM shown in Figure 21. Master Parallel Low and High modes automatically use parallel data supplied to the D0–D7 pins in response to the 16-bit address generated by the LCA device. Figure 22 shows an example of the parallel Master mode connections required. The LCA HEX starting address is 0000 and increments for Master Low mode and it is FFFF and decrements for Master High mode. These two modes provide address compatibility with microprocessors which begin execution from opposite ends of memory.

Peripheral Mode

Peripheral mode provides a simplified interface through which the device may be loaded byte-wide, as a processor peripheral. Figure 23 shows the peripheral mode connections. Processor write cycles are decoded from the common assertion of the active low Write Strobe ($\overline{WS}$), and two active low and one active high Chip Selects ($\overline{CS0}$, $\overline{CS1}$, CS2). The LCA device generates a configuration clock from the internal timing generator and serializes the parallel input data for internal framing or for succeeding slaves on Data Out (DOUT). A output High on READY/$\overline{BUSY}$ pin indicates the completion of loading for each byte when the input register is ready for a new byte. As with Master modes, Peripheral mode may also be used as a lead device for a daisy-chain of slave devices.

Slave Serial Mode

Slave Serial mode provides a simple interface for loading the Logic Cell Array configuration as shown in Figure 24. Serial data is supplied in conjunction with a synchronizing input clock. Most Slave mode applications are in daisy-chain configurations in which the data input is driven from the previous Logic Cell Array's data out, while the clock is supplied by a lead device in Master or Peripheral mode. Data may also be supplied by a processor or other special circuits.

Daisy Chain

The XACT development system is used to create a composite configuration for selected LCA devices including: a preamble, a length count for the total bitstream, multiple concatenated data programs and a postamble plus an additional fill bit per device in the serial chain. After loading and passing-on the preamble and length count to a possible daisy-chain, a lead device will load its configuration data frames while providing a High DOUT to possible down-stream devices as shown in Figure 22. Loading continues while the lead device has received its configuration program and the current length count has not reached the full value. The additional data is passed through the lead device and appears on the Data Out (DOUT) pin in serial form. The lead device also generates the Configuration Clock (CCLK) to synchronize the serial output data and data in of down-stream LCA devices. Data is read in on DIN of slave devices by the positive edge of CCLK and shifted out the DOUT on the negative edge of CCLK. A parallel Master mode device uses its internal timing generator to produce an internal CCLK of 8 times its EPROM address rate, while a Peripheral mode device produces a burst of 8 CCLKs for each chip select and write-strobe cycle. The internal timing generator continues to operate for general timing and synchronization of inputs in all modes.

Special Configuration Functions

The configuration data includes control over several special functions in addition to the normal user logic functions and interconnect.

- Input thresholds
- Readback disable
- DONE pull-up resistor
- DONE timing
- RESET timing
- Oscillator frequency divided by two

Each of these functions is controlled by configuration data bits which are selected as part of the normal XACT development system bitstream generation process.

Input Thresholds

Prior to the completion of configuration all LCA device input thresholds are TTL compatible. Upon completion of configuration, the input thresholds become either TTL or CMOS compatible as programmed. The use of the TTL threshold option requires some additional supply current for threshold shifting. The exception is the threshold of the $\overline{PWRDWN}$ input and direct clocks which always have a CMOS input. Prior to the completion of configuration the user I/O pins each have a high impedance pull-up. The

configuration program can be used to enable the IOB pull-up resistors in the Operational mode to act either as an input load or to avoid a floating input on an otherwise unused pin.

Readback

The contents of a Logic Cell Array may be read back if it has been programmed with a bitstream in which the Readback option has been enabled. Readback may be used for verification of configuration and as a method of determining the state of internal logic nodes during debugging. There are three options in generating the configuration bitstream.

- "Never" inhibits the Readback capability.
- "One-time," inhibits Readback after one Readback has been executed to verify the configuration.
- "On-command" allows unrestricted use of Readback.

Readback is accomplished without the use of any of the user I/O pins; only M0, M1 and CCLK are used. The initiation of Readback is produced by a Low to High transition of the M0/RTRIG (Read Trigger) pin. The CCLK input must then be driven by external logic to read back the configuration data. The first three Low-to-High CCLK transitions clock out dummy data. The subsequent Low-to-High CCLK transitions shift the data frame information out on the M1/RDATA (Read Data) pin. Note that the logic polarity is always inverted, a zero in configuration becomes a one in Readback, and vice versa. Note also that each Readback frame has one Start bit (read back as a one) but, unlike in configuration, each Readback frame has only one Stop bit (read back as a zero). The third leading dummy bit mentioned above can be considered the Start bit of the first frame. All data frames must be read back to complete the process and return the Mode Select and CCLK pins to their normal functions.

Readback data includes the current state of each CLB flip-flop, each input flip-flop or latch, and each device pad. These data are imbedded into unused configuration bit positions during Readback. This state information is used by the XACT development system In-Circuit Verifier to provide visibility into the internal operation of the logic while the system is operating. To readback a uniform time-sample of all storage elements, it may be necessary to inhibit the system clock.

Reprogram

To initiate a re-programming cycle, the dual-function pin DONE/PROG must be given a High-to-Low transition. To reduce sensitivity to noise, the input signal is filtered for two cycles of the LCA device internal timing generator. When reprogram begins, the user-programmable I/O output buffers are disabled and high-impedance pull-ups are provided for the package pins. The device returns to the Clear state and clears the configuration memory before it indicates 'initialized'. Since this Clear operation uses chip-individual internal timing, the master might complete the Clear operation and then start configuration before the slave has completed the Clear operation. To avoid this problem, the slave INIT pins must be AND-wired and used to force a RESET on the master (see Figure 22). Reprogram control is often implemented using an external open-collector driver which pulls DONE/PROG Low. Once a stable request is recognized, theDONE/PROG pin is held Low until the new configuration has been completed. Even if the re-program request is externally held Low beyond the configuration period, the LCA device will begin operation upon completion of configuration.

DONE Pull-up

DONE/PROG is an open-drain I/O pin that indicates the LCA device is in the operational state. An optional internal pull-up resistor can be enabled by the user of the XACT development system when MAKEBITS is executed. The DONE/PROG pins of multiple LCA devices in a daisy-chain may be connected together to indicate all are DONE or to direct them all to reprogram.

DONE Timing

The timing of the DONE status signal can be controlled by a selection in the MakeBits program to occur either a CCLK cycle before, or after, the outputs going active. See Figure 20. This facilitates control of external functions such as a PROM enable or holding a system in a wait state.

RESET Timing

As with DONE timing, the timing of the release of the internal reset can be controlled by a selection in the MakeBits program to occur either a CCLK cycle before, or after, the outputs going active. See Figure 20. This reset keeps all user programmable flip-flops and latches in a zero state during configuration.

Crystal Oscillator Division

A selection in the MakeBits program allows the user to incorporate a dedicated divide-by-two flip-flop between the crystal oscillator and the alternate clock line. This guarantees a symmetrical clock signal. Although the frequency stability of a crystal oscillator is very good, the symmetry of its waveform can be affected by bias or feedback drive.

The following seven pages describe the different configuration modes in detail

Master Serial Mode

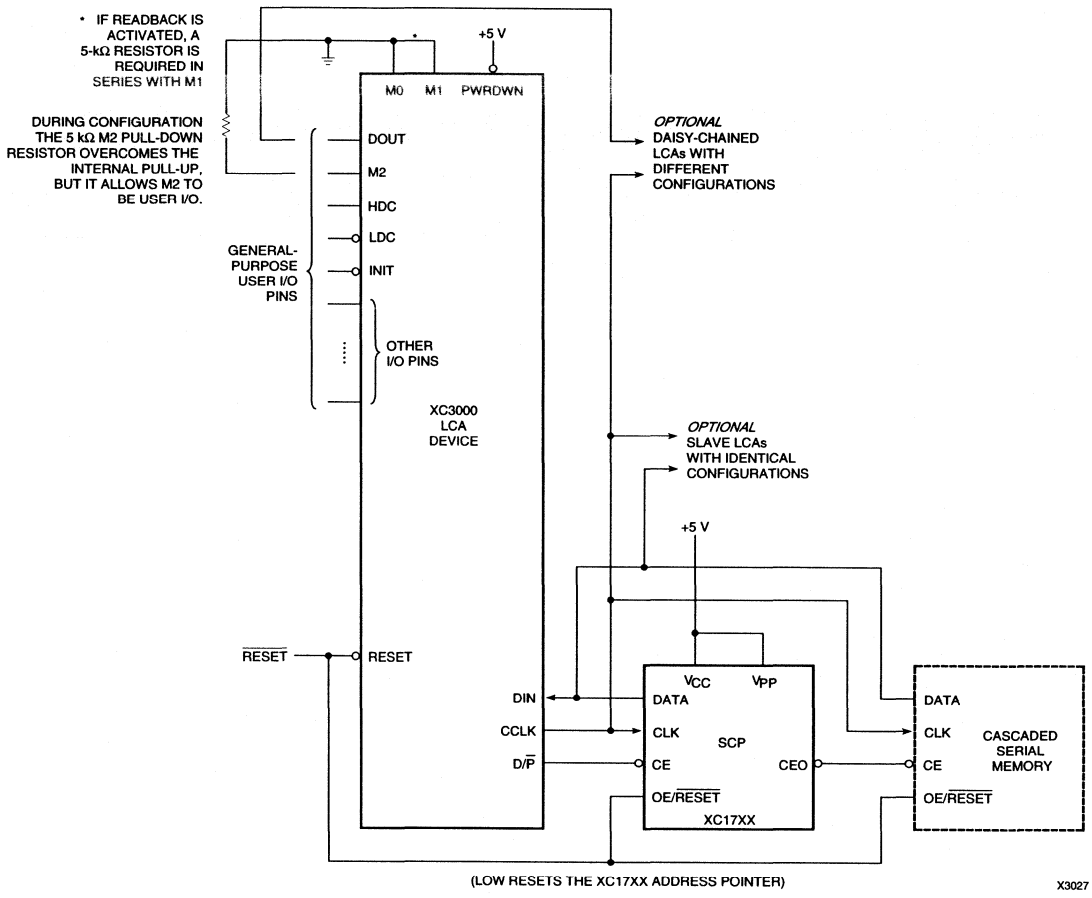


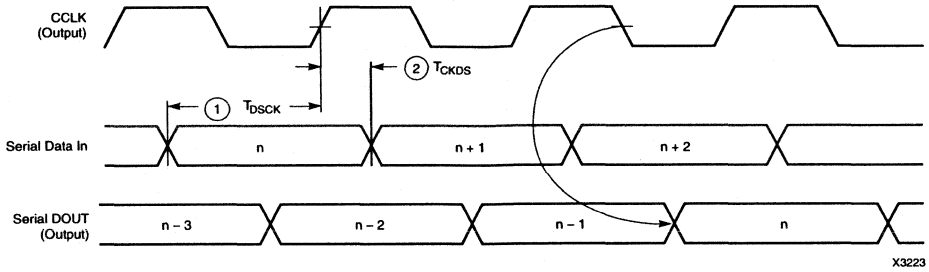
Figure 21. Master Serial Mode

In Master Serial mode, the CCLK output of the lead LCA device drives a Xilinx Serial PROM that feeds the LCA DIN input. Each rising edge of the CCLK output increments the Serial PROM internal address counter. This puts the next data bit on the SPROM data output, connected to the LCA DIN pin. The lead LCA device accepts this data on the subsequent rising CCLK edge.

The lead LCA device then presents the preamble data (and all data that overflows the lead device) on its DOUT pin. There is an internal delay of 1.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

The SPROM CE input can be driven from either $\overline{LDC}$ or DONE . Using $\overline{LDC}$ avoids potential contention on the DIN pin, if this pin is configured as user-I/O, but $\overline{LDC}$ is then restricted to be a permanently High user output. Using DONE also avoids contention on DIN, provided the early DONE option is invoked.

Master Serial Mode Programming Switching Characteristics



	Speed Grade		Min	Max	Units
	Description	Symbol			
CCLK	Data In setup	1 T_{DSCK}	60		ns
	Data In hold	2 $_{CKDS}$	0		ns

Notes:
1. At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until V_{CC} has reached 4.0 V (2.5 V for the XC3000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require >6-µs High level on RESET, followed by a >6-µs Low level on $\overline{\text{RESET}}$ and D/$\overline{\text{P}}$ after V_{CC} has reached 4.0 V (2.5 V for the XC3000L).
2. Configuration can be controlled by holding $\overline{\text{RESET}}$ Low with or until after the $\overline{\text{INIT}}$ of all daisy-chain slave-mode devices is High.
3. Master-serial-mode timing is based on slave-mode testing.

Master Parallel Mode

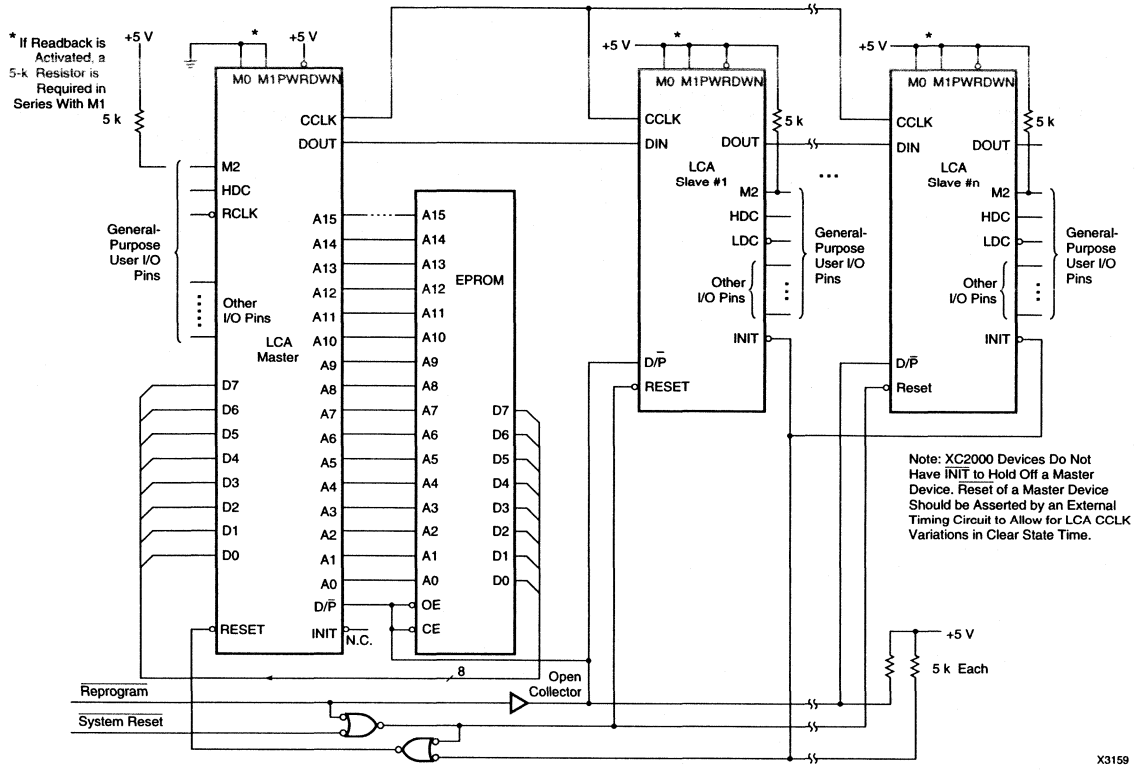
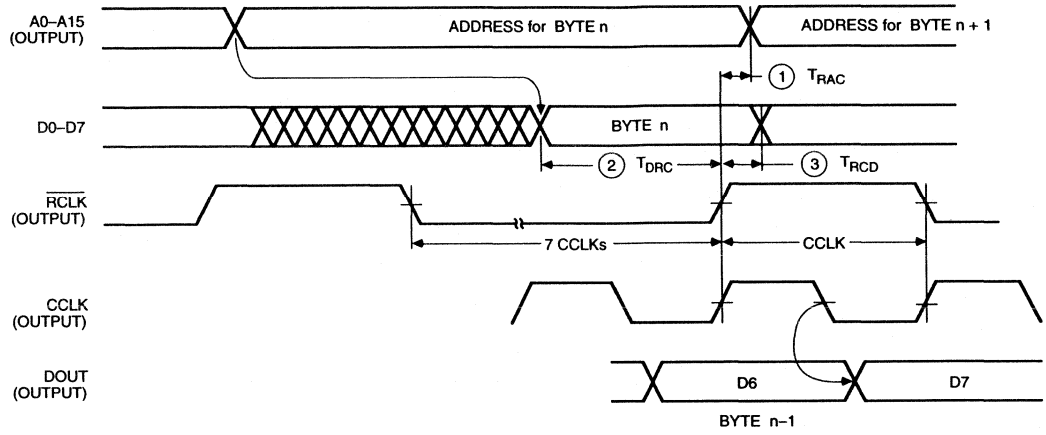


Figure 22. Master Parallel Mode

In Master Parallel mode, the lead LCA device directly addresses an industry-standard byte-wide EPROM and accepts eight data bits right before incrementing (or decrementing) the address outputs.

The eight data bits are serialized in the lead LCA device, which then presents the preamble data (and all data that overflows the lead device) on the DOUT pin. There is an internal delay of 1.5 CCLK periods, after the rising CCLK edge that accepts a byte of data, and also changes the EPROM address, until the falling CCLK edge that makes the LSB (D0) of this byte appear at DOUT. This means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy chain accepts data on the subsequent rising CCLK edge.

Master Parallel Mode Programming Switching Characteristics



	Description	Symbol		Min	Max	Units
RCLK	To address valid	1	T_{RAC}	0	200	ns
	To data setup	2	T_{DRC}	60		ns
	To data hold	3	T_{RCD}	0		ns
	RCLK High		T_{RCH}	600		ns
	RCLK Low		T_{RCL}	4.0		µs

Notes: 1. At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until V_{CC} has reached 4.0 V (2.5 V for the XC3000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require a >6-µs High level on $\overline{RESET}$, followed by a >6-µs Low level on $\overline{RESET}$ and D/P after V_{CC} has reached 4.0 V (2.5 V for the XC3000L).

2. Configuration can be controlled by holding $\overline{RESET}$ Low with or until after the $\overline{INIT}$ of all daisy-chain slave-mode devices is High.

This timing diagram shows that the EPROM requirements are extremely relaxed:
EPROM access time can be longer than 4000 ns. EPROM data output has no hold time requirements.

Peripheral Mode

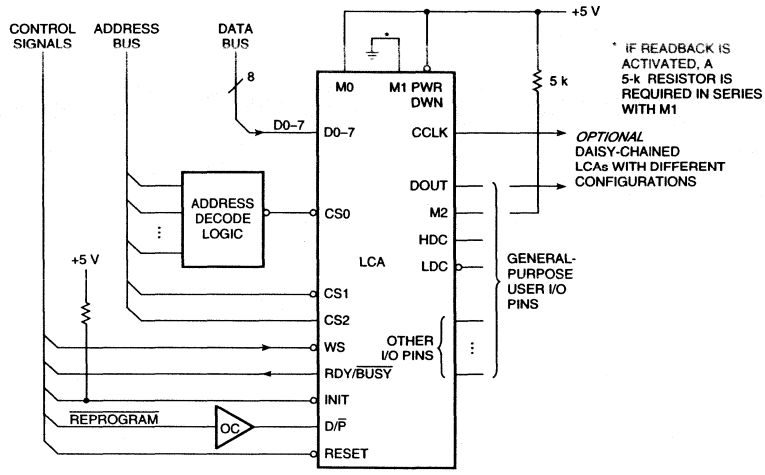


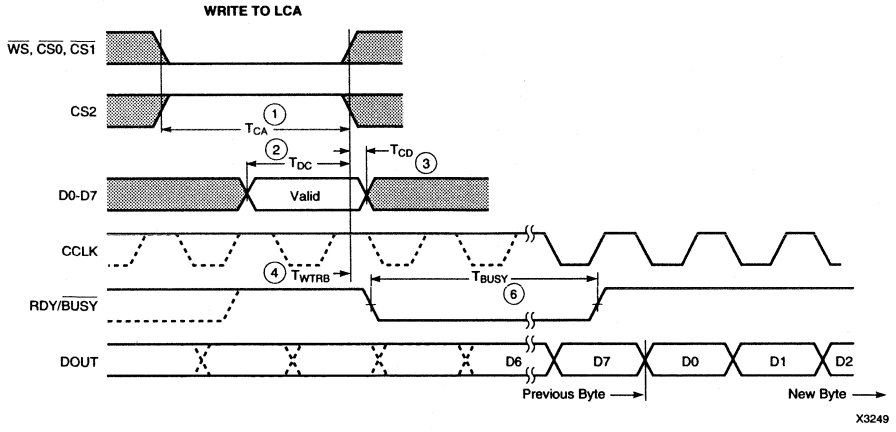
Figure 23. Peripheral Mode.

Peripheral mode uses the trailing edge of the logic AND condition of the $\overline{CS0}$, $\overline{CS1}$, CS2, and $\overline{WS}$ inputs to accept byte-wide data from a microprocessor bus. In the lead LCA device, this data is loaded into a double-buffered UART-like parallel-to-serial converter and is serially shifted into the internal logic. The lead LCA device presents the preamble data (and all data that overflows the lead device) on the DOUT pin.

The Ready/Busy output from the lead LCA device acts as a handshake signal to the microprocessor. RDY/$\overline{BUSY}$ goes Low when a byte has been received, and goes High again when the byte-wide input buffer has transferred its information into the shift register, and the buffer is ready to receive new data. The length of the $\overline{BUSY}$ signal depends on the activity in the UART. If the shift register had been empty when the new byte was received, the $\overline{BUSY}$ signal lasts for only two CCLK periods. If the shift register was still full when the new byte was received, the $\overline{BUSY}$ signal can be as long as nine CCLK periods.

Note that after the last byte has been entered, only seven of its bits are shifted out. CCLK remains High with DOUT equal to bit 6 (the next-to-last bit) of the last byte entered.

Peripheral Mode Programming Switching Characteristics



	Description	Symbol		Min	Max	Units
Write	Effective Write time required (Assertion of $\overline{CS0}$, $\overline{CS1}$, CS2, $\overline{WS}$)	1	T_{CA}	100		ns
	DIN Setup time required	2	T_{DC}	60		ns
	DIN Hold time required	3	T_{CD}	0		ns
	RDY/$\overline{BUSY}$ delay after end of $\overline{WS}$	4	T_{WTRB}		60	ns
RDY	Earliest next $\overline{WS}$ after end of $\overline{BUSY}$	5	T_{RBWT}	0		ns
	$\overline{BUSY}$ Low time generated	6	T_{BUSY}	2.5	9	CCLK Periods

Notes:

1. At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until V_{CC} has reached 4.0 V (2.5 V for the XC3000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require a >6-µs High level on $\overline{RESET}$, followed by a >6-µs Low level on $\overline{RESET}$ and D/$\overline{P}$ after V_{CC} has reached 4.0 V (2.5 V for the XC3000L).
2. Configuration must be delayed until the $\overline{INIT}$ of all LCAs is High.
3. Time from end of $\overline{WS}$ to CCLK cycle for the new byte of data depends on completion of previous byte processing and the phase of the internal timing generator for CCLK.
4. CCLK and DOUT timing is tested in slave mode.
5. T_{BUSY} indicates that the double-buffered parallel-to-serial converter is not yet ready to receive new data. The shortest T_{BUSY} occurs when a byte is loaded into an empty parallel-to-serial converter. The longest T_{BUSY} occurs when a new word is loaded into the input register before the second-level buffer has started shifting out data.

This timing diagram shows very relaxed requirements: Data need not be held beyond the rising edge of $\overline{WS}$. $\overline{BUSY}$ will go active within 60 ns after the end of $\overline{WS}$. $\overline{BUSY}$ will stay active for several microseconds. $\overline{WS}$ may be asserted immediately after the end of $\overline{BUSY}$.

Slave Serial Mode

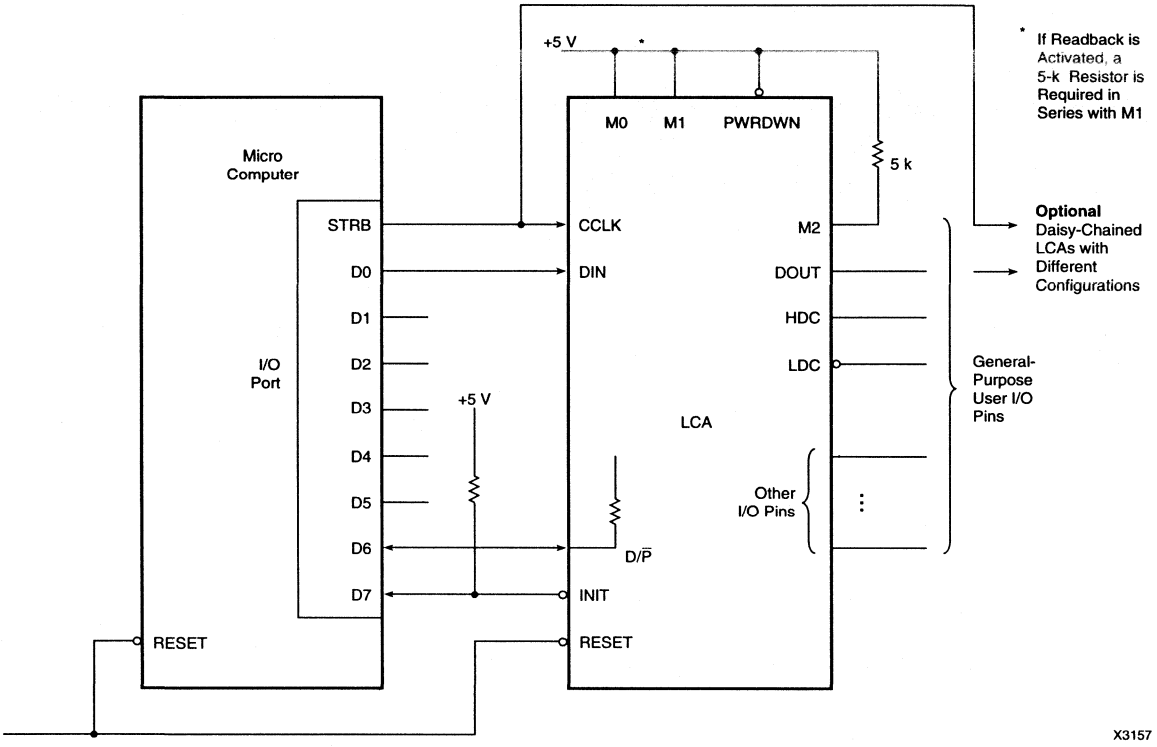
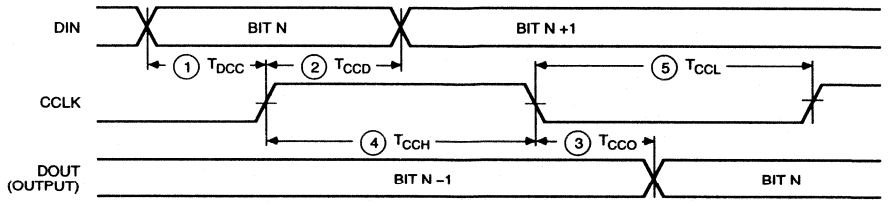


Figure 24. Slave Serial Mode.

In Slave Serial mode, an external signal drives the CCLK input(s) of the LCA device(s). The serial configuration bitstream must be available at the DIN input of the lead LCA device a short set-up time before each rising CCLK edge. The lead LCA device then presents the preamble data (and all data that overflows the lead device) on its DOUT pin. There is an internal delay of 0.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

Slave Serial Mode Programming Switching Characteristics

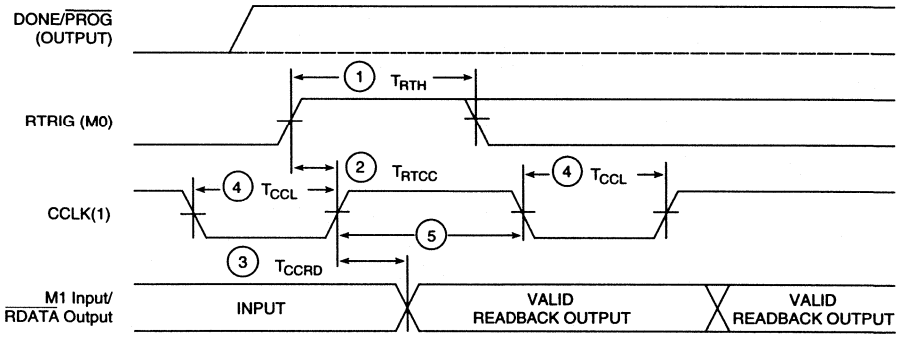


1105 31

	Description	Symbol		Min	Max	Units
CCLK	To DOUT	3	T_{CCO}		100	ns
	DIN setup	1	T_{DCC}	60		ns
	DIN hold	2	T_{CCD}	0		ns
	High time	4	T_{CCH}	0.05		µs
	Low time (Note 1)	5	T_{CCL}	0.05	5.0	µs
	Frequency		F_{CC}		10	MHz

Notes: 1. The max limit of CCLK Low time is caused by dynamic circuitry inside the LCA device.
2. Configuration must be delayed until the $\overline{INIT}$ of all LCA devices is High.
3. At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until V_{CC} has reached 4.0 V (2.5 V for the XC3000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require a >6-µs High level on $\overline{RESET}$, followed by a >6-µs Low level on $\overline{RESET}$ and D/$\overline{P}$ after V_{CC} has reached 4.0 V (2.5 V for the XC3000L).

Program Readback Switching Characterisctics

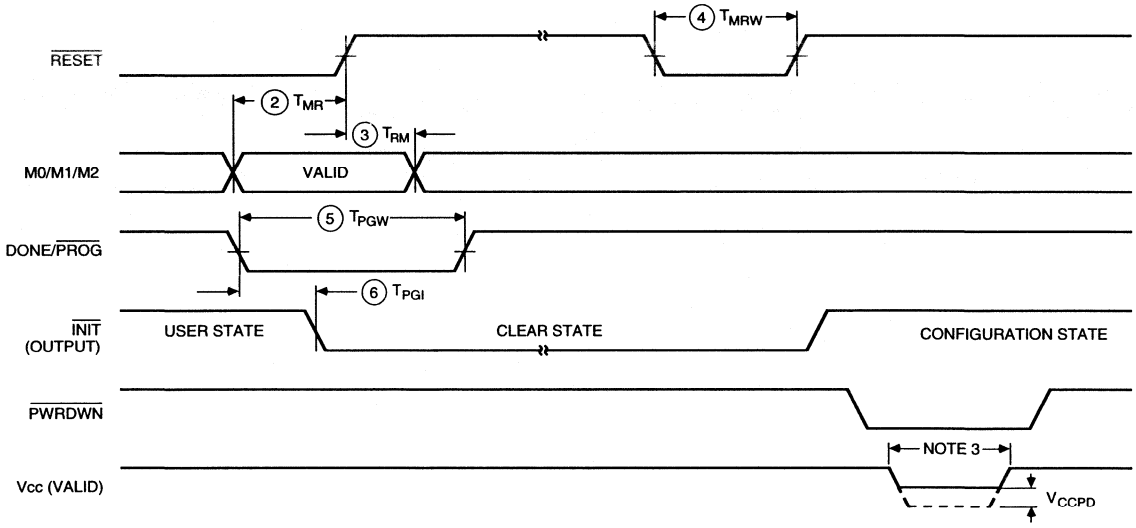


X4395

	Description	Symbol		Min	Max	Units
RTRIG	RTRIG High	1	T_{RTH}	250		ns
CCLK	RTRIG setup	2	T_{RTCC}	200		ns
	RDATA delay	3	T_{CCRD}		100	ns
	High time	5	T_{CCHR}	0.5		µs
	Low time	4	T_{CCLR}	0.5	5	µs

Notes: 1. During Readback, CCLK frequency may not exceed 1 MHz.
2. RETRIG (M0 positive transition) shall not be done until after one clock following active I/O pins.
3. Readback should not be initiated until configuration is complete.
4. T_{CCLR} is 5 µs min to 15 µs max for XC3000L.

General LCA Switching Characteristics



Description		Symbol		Min	Max	Units
RESET (2)	M0, M1, M2 setup time required	2	T_{MR}	1		µs
	M0, M1, M2 hold time required	3	T_{RM}	3		µs
	RESET Width (Low) req. for Abort	4	T_{MRW}	6		µs
DONE/PROG	Width (Low) required for Re-config.	5	T_{PGW}	6		µs
	INIT response after D/P is pulled Low	6	T_{PGI}		7	µs
PWRDWN (3)	Power Down V_{CC}		V_{CCPD}	2.3		V

Notes: 1. At power-up, V_{cc} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until V_{cc} has reached 4.0 V (2.5 V for XC3000L). A very long V_{cc} rise time of >100 ms, or a non-monotonically rising V_{cc} may require a >1-µs High level on RESET, followed by a >6-µs Low level on RESET and D/P after V_{cc} has reached 4.0 V (2.5 V for XC3000L).

2. RESET timing relative to valid mode lines (M0, M1, M2) is relevant when RESET is used to delay configuration. The specified hold time is caused by a shift-register filter slowing down the response to RESET during configuration.

3. PWRDWN transitions must occur while V_{cc} >4.0 V(2.5 V for XC3000L).

Device Performance

The XC3000 families of FPGAs can achieve very high performance. This is the result of

- A sub-micron manufacturing process, developed and continuously being enhanced for the production of state-of-the-art CMOS SRAMs.
- Careful optimization of transistor geometries, circuit design, and lay-out, based on years of experience with the XC3000 family.
- A look-up table based, coarse-grained architecture that can collapse multiple-layer combinatorial logic into a single function generator. One CLB can implement up to four layers of conventional logic in as little as 2.7 ns.

Actual system performance is determined by the timing of critical paths, including the delay through the combinatorial and sequential logic elements within CLBs and IOBs, plus the delay in the interconnect routing. The ac-timing specifications state the worst-case timing parameters for the various logic resources available in the XC3000-families architecture. Figure 25 shows a variety of elements involved in determining system performance.

Logic block performance is expressed as the propagation time from the interconnect point at the input to the block to the output of the block in the interconnect area. Since combinatorial logic is implemented with a memory lookup table within a CLB, the combinatorial delay through the CLB, called T_{ILO}, is always the same, regardless of the function being implemented. For the combinatorial logic function driving the data input of the storage element, the critical timing is data set-up relative to the clock edge provided to the flip-flop element. The delay from the clock source to the output of the logic block is critical in the timing signals produced by storage elements. Loading of a logic-block output is limited only by the resulting propagation delay of the larger interconnect network. Speed performance of the logic block is a function of supply voltage and temperature. See Figure 26.

Interconnect performance depends on the routing resources used to implement the signal path. Direct interconnects to the neighboring CLB provide an extremely fast path. Local interconnects go through switch matrices (magic boxes) and suffer an RC delay, equal to the resistance of the pass transistor multiplied by the capacitance of the driven metal line. Longlines carry the signal across the length or breadth of the chip with only one access delay. Generous on-chip signal buffering makes performance relatively insensitive to signal fan-out; increasing fan-out from 1 to 8 changes the CLB delay by only 10%. Clocks can be distributed with two low-skew clock distribution networks.

The tools in the XACT Development System used to place and route a design in an XC3000 FPGA (the Automatic Place and Route [APR] program and the XACT Design Editor)automatically calculate the actual maximum worst-case delays along each signal path. This timing information can be back-annotated to the design's netlist for use in timing simulation or examined with X-DELAY, a static timing analyzer.

Actual system performance is applications dependent. The maximum clock rate that can be used in a system is determined by the critical path delays within that system. These delays are combinations of incremental logic and routing delays, and vary from design to design. In a synchronous system, the maximum clock rate depends on the number of combinatorial logic layers between re-synchronizing flip-flops. Figure 27 shows the achievable clock rate as a function of the number of CLB layers.

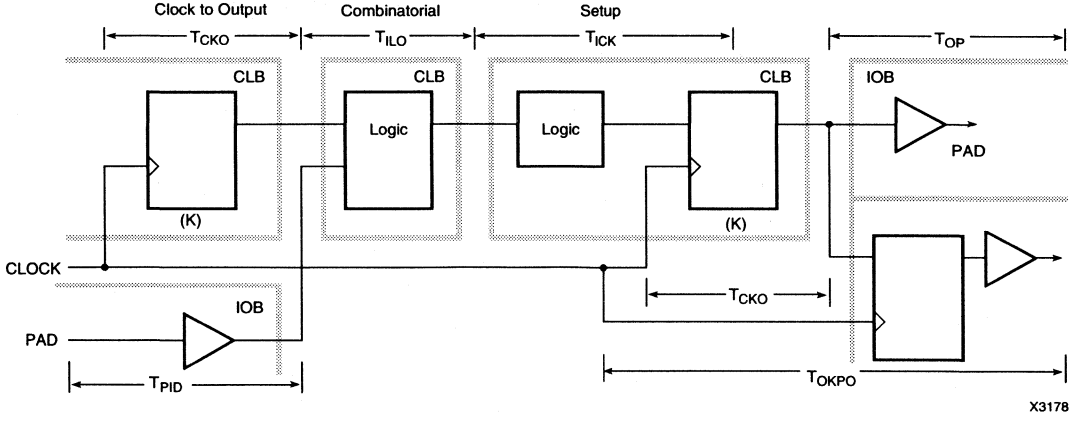


Figure 25. Primary Block Speed Factors. Actual timing is a function of various block factors combined with routing factors. Overall performance can be evaluated with the XACT timing calculator or by an optional simulation.

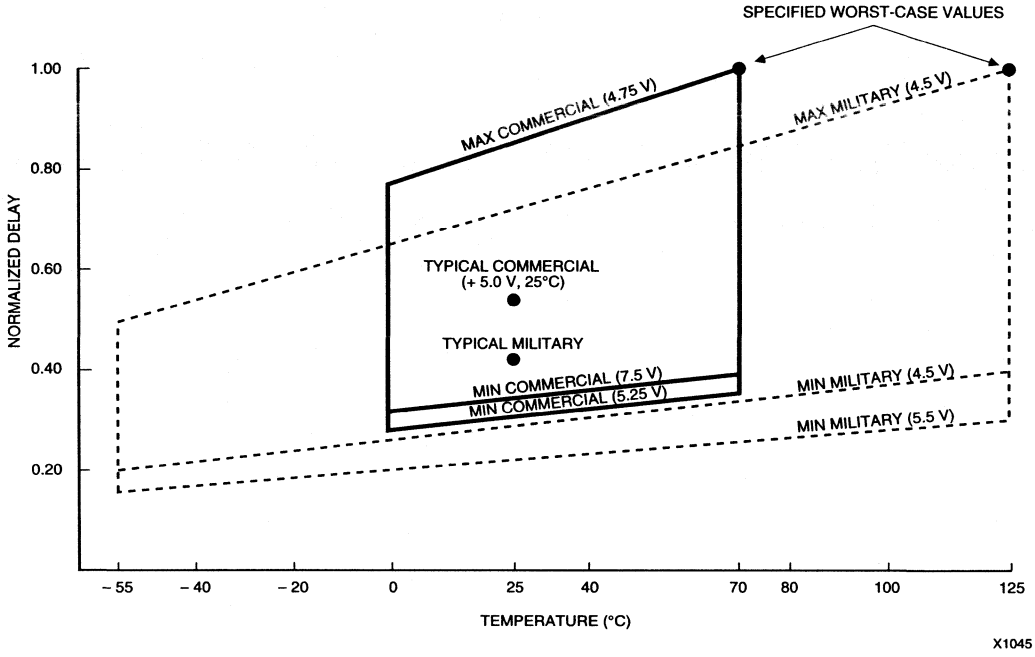


Figure 26. Relative Delay as a Function of Temperature, Supply Voltage and Processing Variations

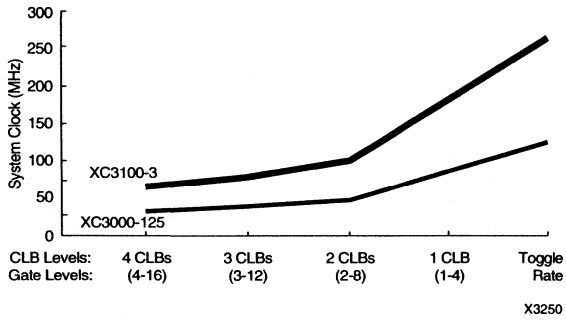


Figure 27. Clock Rate as a Function of Logic Complexity (Number of Combinational Levels between Flip-Flops)

Power

Power Distribution

Power for the LCA device is distributed through a grid to achieve high noise immunity and isolation between logic and I/O. Inside the LCA device, a dedicated V_{CC} and ground ring surrounding the logic array provides power to the I/O drivers. An independent matrix of V_{CC} and groundlines supplies the interior logic of the device. This power distribution grid provides a stable supply and ground for all internal logic, providing the external package power pins are all connected and appropriately decoupled. Typically a 0.1-µF capacitor connected near the V_{CC} and ground pins will provide adequate decoupling.

Output buffers capable of driving the specified 4- or 8-mA loads under worst-case conditions may be capable of driving as much as 25 to 30 times that current in a best case. Noise can be reduced by minimizing external load capacitance and reducing simultaneous output transitions in the same direction. It may also be beneficial to locate heavily loaded output buffers near the ground pads. The I/O Block output buffers have a slew-limited mode which should be used where output rise and fall times are not speed critical. Slew-limited outputs maintain their dc drive capability, but generate less external reflections and internal noise.

Dynamic Power Consumption

	XC3042	XC3042A	XC3042L	XC3142A	
One CLB driving three local interconnects	0.25	0.17	0.07	0.25	mW per MHz
One global clock buffer and clock line	2.25	1.40	0.50	1.70	mW per MHz
One device output with a 50 pF load	1.25	1.25	0.55	1.25	mW per MHz

Power Consumption

The Logic Cell Array exhibits the low power consumption characteristic of CMOS ICs. For any design, the configuration option of TTL chip input threshold requires power for the threshold reference. The power required by the static memory cells that hold the configuration data is very low and may be maintained in a power-down mode.

Typically, most of power dissipation is produced by external capacitive loads on the output buffers. This load and frequency dependent power is 25 μW/pF/MHz per output. Another component of I/O power is the external dc loading on all output pins.

Internal power dissipation is a function of the number and size of the nodes, and the frequency at which they change. In an LCA device, the fraction of nodes changing on a given clock is typically low (10-20%). For example, in a long binary counter, the total activity of all counter flip-flops is equivalent to that of only two CLB outputs toggling at the clock frequency. Typical global clock-buffer power is between 2.0 mW/MHz for the XC3020 and 3.5 mW/MHz for the XC3090. The internal capacitive load is more a function of interconnect than fan-out. With a typical load of three general interconnect segments, each CLB output requires about 0.25 mW per MHz of its output frequency.

Because the control storage of the Logic Cell Array is CMOS static memory, its cells require a very low standby current for data retention. In some systems, this low data retention current characteristic can be used as a method of preserving configurations in the event of a primary power loss. The Logic Cell Array has built in Powerdown logic which, when activated, will disable normal operation of the device and retain only the configuration data. All internal operation is suspended and output buffers are placed in their high-impedance state with no pull-ups. Different from the XC3000 family which can be powered down to a current consumption of a few microamps, the XC3100 draws 5 mA, even in power-down. This makes power-down operation less meaningful. In contrast, I_{CCPD} for the XC3000L is only 10 μA.

To force the Logic Cell Array into the Powerdown state, the user must pull the $\overline{\text{PWRDWN}}$ pin Low and continue to supply a retention voltage to the V_{CC} pins. When normal power is restored, V_{CC} is elevated to its normal operating voltage and $\overline{\text{PWRDWN}}$ is returned to a High. The Logic Cell Array resumes operation with the same internal sequence that occurs at the conclusion of configuration. Internal-I/O and logic-block storage elements will be reset, the outputs will become enabled and the DONE/$\overline{\text{PROG}}$ pin will be released.

When V_{CC} is shut down or disconnected, some power might unintentionally be supplied from an incoming signal driving an I/O pin. The conventional electro-static input protection is implemented with diodes to the supply and ground. A positive voltage applied to an input (or output) will cause the positive protection diode to conduct and drive the V_{CC} connection. This condition can produce invalid power conditions and should be avoided. A large series resistor might be used to limit the current or a bipolar buffer may be used to isolate the input signal.

Pin Descriptions

Permanently Dedicated Pins.

V_{CC}
Two to eight (depending on package type) connections to the positive V supply voltage. All must be connected.

GND
Two to eight (depending on package type) connections to ground. All must be connected.

PWRDWN
A Low on this CMOS-compatible input stops all internal activity, but retains configuration. All flip-flops and latches are reset, all outputs are 3-stated, and all inputs are interpreted as High, independent of their actual level. When PWDWN returns High, the LCA device becomes operational with DONE Low for two cycles of the internal 1-MHz clock.Before and during configuration, PWRDWN must be High. If not used, PWRDWN must be tied to V_{CC}.

RESET
This is an active Low input which has three functions.

Prior to the start of configuration, a Low input will delay the start of the configuration process. An internal circuit senses the application of power and begins a minimal time-out cycle. When the time-out and RESET are complete, the levels of the M lines are sampled and configuration begins.

If RESET is asserted during a configuration, the LCA device is re-initialized and restarts the configuration at the termination of RESET.

If RESET is asserted after configuration is complete, it provides a global asynchronous RESET of all IOB and CLB storage elements of the LCA device.

CCLK
During configuration, Configuration Clock is an output of an LCA device in Master mode or Peripheral mode, but an input in Slave mode. During Readback, CCLK is a clock input for shifting configuration data out of the LCA device

CCLK drives dynamic circuitry inside the LCA device. The Low time may, therefore, not exceed a few microseconds. When used as an input, CCLK must be "parked High". An internal pull-up resistor maintains High when the pin is not being driven.

DONE/PROG (D/P)
DONE is an open-drain output, configurable with or without an internal pull-up resistor. At the completion of configuration, the LCA device circuitry becomes active in a synchronous order; DONE is programmed to go active High one cycle either before or after the outputs go active.

Once configuration is done, a High-to-Low transition of this pin will cause an initialization of the LCA device and start a reconfiguration.

M0/RTRIG
As Mode 0, this input is sampled on power-on to determine the power-on delay (2^{14} cycles if M0 is High, 2^{16} cycles if M0 is Low). Before the start of configuration, this input is again sampled together with M1, M2 to determine the configuration mode to be used .

A Low-to-High input transition, after configuration is complete, acts as a Read Trigger and initiates a Readback of configuration and storage-element data clocked by CCLK. By selecting the appropriate Readback option when generating the bitstream, this operation may be limited to a single Readback, or be inhibited altogether.

M1/RDATA
As Mode 1, this input and M0, M2 are sampled before the start of configuration to establish the configuration mode to be used. If Readback is never used, M1 can be tied directly to ground or V_{CC}. If Readback is ever used, M1 must use a 5-kΩ resistor to ground or V_{CC}, to accommodate the RDATA output.

As an active-Low Read Data, after configuration is complete, this pin is the output of the Readback data.

User I/O Pins that can have special functions.

M2
During configuration, this input has a weak pull-up resistor. Together with M0 and M1, it is sampled before the start of configuration to establish the configuration mode to be used. After configuration, this pin is a user-programmable I/O pin.

HDC
During configuration, this output is held at a High level to indicate that configuration is not yet complete. After configuration, this pin is a user-programmable I/O pin.

$\overline{LDC}$
During Configuration, this output is held at a Low level to indicate that the configuration is not yet complete. After configuration, this pin is a user-programmable I/O pin. $\overline{LDC}$ is particularly useful in Master mode as a Low enable for an EPROM, but it must then be programmed as a High after configuration.

$\overline{INIT}$
This is an active Low open-drain output with a weak pull-up and is held Low during the power stabilization and internal clearing of the configuration memory. It can be used to indicate status to a configuring microprocessor or, as a wired AND of several slave mode devices, a hold-off signal for a master mode device. After configuration this pin becomes a user-programmable I/O pin.

BCLKIN
This is a direct CMOS level input to the alternate clock buffer (Auxiliary Buffer) in the lower right corner.

XTL1
This user I/O pin can be used to operate as the output of an amplifier driving an external crystal and bias circuitry.

XTL2
This user I/O pin can be used as the input of an amplifier connected to an external crystal and bias circuitry. The I/O Block is left unconfigured. The oscillator configuration is activated by routing a net from the oscillator buffer symbol output and by the MakeBits program.

$\overline{CS0}$, $\overline{CS1}$, CS2, $\overline{WS}$
These four inputs represent a set of signals, three active Low and one active High, that are used to control configuration-data entry in the Peripheral mode. Simultaneous assertion of all four inputs generates a Write to the internal data buffer. The removal of any assertion clocks in the D0-D7 data. In Master-Parallel mode, WS and CS2 are the A0 and A1 outputs. After configuration, these pins are user-programmable I/O pins.

RDY/$\overline{BUSY}$
During Peripheral Parallel mode configuration this pin indicates when the chip is ready for another byte of data to be written to it. After configuration is complete, this pin becomes a user-programmed I/O pin.

$\overline{RCLK}$
During Master Parallel mode configuration, each change on the A0-15 outputs is preceded by a rising edge on $\overline{RCLK}$, a redundant output signal. After configuration is complete, this pin becomes a user-programmed I/O pin.

D0-D7
This set of eight pins represents the parallel configuration byte for the parallel Master and Peripheral modes. After configuration is complete, they are user-programmed I/O pins.

A0-A15
During Master Parallel mode, these 16 pins present an address output for a configuration EPROM. After configuration, they are user-programmable I/O pins.

DIN
During Slave or Master Serial configuration, this pin is used as a serial-data input. In the Master or Peripheral configuration, this is the Data 0 input. After configuration is complete, this pin becomes a user-programmed I/O pin.

DOUT
During configuration this pin is used to output serial-configuration data to the DIN pin of a daisy-chained slave. After configuration is complete, this pin becomes a user-programmed I/O pin.

TCLKIN
This is a direct CMOS-level input to the global clock buffer. This pin can also be configured as a user programmable I/O pin. However, since TCLKIN is the preferred input to the global clock net, and the global clock net should be used as the primary clock source, this pin is usually the clock input to the chip.

Unrestricted User I/O Pins.

I/O
An I/O pin may be programmed by the user to be an Input or an Output pin following configuration. All unrestricted I/O pins, plus the special pins mentioned on the following page, have a weak pull-up resistor of 50 kΩ to 100 kΩ that becomes active as soon as the device powers up, and stays active until the end of configuration.

Before and during configuration, all outputs that are not used for the configuration process are 3-stated with a 50 kΩ to 100 kΩ pull-up resistor.

Pin Functions During Configuration

Configuration Mode <M2:M1:M0>					***									****	
SLAVE <1:1:1>	MASTER-SER <0:0:0>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>	44 PLCC	68 PLCC	84 PLCC	84 PGA	100 PQFP	100 TQFP	132 PGA	160 PQFP	175 PGA	208 PQFP	User Operation
PWRDWN (I)	PWRDWN (I)	PWRDWN (I)	PWRDWN (I)	PWRDWN (I)	7	10	12	B2	29	26	A1	159	B2	3	PWRDWN (I)
VCC	VCC	VCC	VCC	VCC	12	18	22	F3	41	38	C8	20	D9	26	VCC
M1 (HIGH) (I)	M1 (LOW) (I)	M1 (LOW) (I)	M1 (HIGH) (I)	M1 (LOW) (I)	16	25	31	J2	52	49	B13	40	B14	48	RDATA
M0 (HIGH) (I)	M0 (LOW) (I)	M0 (HIGH) (I)	M0 (LOW) (I)	M0 (LOW) (I)	17	26	32	L1	54	51	A14	42	B15	50	RTRIG (I)
M2 (HIGH) (I)	M2 (LOW) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	M2 (HIGH) (I)	18	27	33	K2	56	53	C13	44	C15	56	I/O
HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	HDC (HIGH)	19	28	34	K3	57	54	B14	45	E14	57	I/O
LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	LDC (LOW)	20	30	36	L3	59	56	D14	49	D16	61	I/O
INIT*	INIT*	INIT*	INIT*	INIT*	22	34	42	K6	65	62	G14	59	H15	77	I/O
GND	GND	GND	GND	GND	23	35	43	J6	66	63	H12	19	J14	79	GND
					26	43	53	L11	76	73	M13	76	P15	100	XTL2 OR I/O
RESET (I)	RESET (I)	RESET (I)	RESET (I)	RESET (I)	27	44	54	K10	78	75	P14	78	R15	102	RESET (I)
DONE	DONE	DONE	DONE	DONE	28	45	55	J10	80	77	N13	80	R14	107	PROGRAM (I)
		DATA 7 (I)	DATA 7 (I)	DATA 7 (I)		46	56	K11	81	78	M12	81	N13	109	I/O
					30	47	57	J11	82	79	P13	82	T14	110	XTL1 OR I/O
		DATA 6 (I)	DATA 6 (I)	DATA 6 (I)		48	58	H10	83	80	N11	86	P12	115	I/O
		DATA 5 (I)	DATA 5 (I)	DATA 5 (I)		49	60	F10	87	84	M9	92	T11	122	I/O
		CS0 (I)				50	61	G10	88	85	N9	93	R10	123	I/O
		DATA 4 (I)	DATA 4 (I)	DATA 4 (I)		51	62	G11	89	86	N8	98	R9	128	I/O
VCC	VCC	VCC	VCC	VCC	34	52	64	F9	91	88	M8	100	N9	130	Vcc
		DATA 3 (I)	DATA 3 (I)	DATA 3 (I)		53	65	F11	92	89	N7	102	P8	132	I/O
		CS1 (I)				54	66	E11	93	90	P6	103	R8	133	I/O
		DATA 2 (I)	DATA 2 (I)	DATA 2 (I)		55	67	E10	94	91	M6	108	R7	138	I/O
		DATA 1 (I)	DATA 1 (I)	DATA 1 (I)		56	70	D10	98	95	M5	114	R5	145	I/O
		RDY/BUSY	RCLK	RCLK		57	71	C11	99	96	N4	115	P5	146	I/O
DIN (I)	DIN (I)	DATA 0 (I)	DATA 0 (I)	DATA 0 (I)	38	58	72	B11	100	97	N2	119	R3	151	I/O
DOUT	DOUT	DOUT	DOUT	DOUT	39	59	73	C10	1	98	M3	120	N4	152	I/O
CCLK (I)	CCLK(O)	CCLK(O)	CCLK(O)	CCLK(O)	40	60	74	A11	2	99	P1	121	R2	153	CCLK (I)
		WS (I)	A0	A0		61	75	B10	5	2	M2	124	P2	161	I/O
		CS2 (I)	A1	A1		62	76	B9	6	3	N1	125	M3	162	I/O
			A2	A2		63	77	A10	8	5	L2	128	P1	165	I/O
			A3	A3		64	78	A9	9	6	L1	129	N1	166	I/O
			A15	A15		65	81	B6	12	9	K1	132	M1	172	I/O
			A4	A4		66	82	B7	13	10	J2	133	L2	173	I/O
			A14	A14		67	83	A7	14	11	H1	136	K2	178	I/O
			A5	A5		68	84	C7	15	12	H2	137	K1	179	I/O
GND	GND	GND	GND	GND	1	1	1	C6	16	13	H3	139	J3	182	GND
			A13	A13		2	2	A6	17	14	G2	141	H2	184	I/O
			A6	A6		3	3	A5	18	15	G1	142	H1	185	I/O
			A12	A12		4	4	B5	19	16	F2	147	F2	192	I/O
			A7	A7		5	5	C5	20	17	E1	148	E1	193	I/O
			A11	A11		6	8	A3	23	20	D1	151	D1	199	I/O
			A8	A8		7	9	A2	24	21	D2	152	C1	200	I/O
			A10	A10		8	10	B3	25	22	B1	155	E3	203	I/O
			A9	A9		9	11	A1	26	26	C2	156	C2	204	I/O
															All Others
						X	X	X	X						XC3020 etc.
					X	X	X	X	X	X					XC3030 etc.
							X	X	X	X	X				XC3042 etc.
							X**				X				XC3064 etc.
							X**					X	X	X	XC3090 etc.
							X**					X	X	X	XC3195

X5266

Represents a 50-kΩ to 100-kΩ pull-up before and during configuration

* INIT is an open drain output during configuration

(I) Represents an input

** Pin assignmnent for the XC3064/XC3090 and XC3195 differ from those shown. See page 2-138.

*** Peripheral mode and master parallel mode are not supported in the PC44 package. See page 2-135.

**** Pin assignments for the XC3195 PQ208 differ from those shown. See page 2-146.

Pin assignments of PGA Footprint PLCC sockets and PGA packages are not electrically identical.

Generic I/O pins are not shown.

The information on this page is provided as a convenient summary. For detailed pin descriptions, see the preceding two pages.

For a detailed description of the configuration modes, see pages 2-190 through 2-200.

For pinout details, see pages 2-136 through 2-146.

Before and during configuration, all outputs that are not used for the configuration process are 3-stated with a 50 kΩ to 100 kΩ pull-up resistor.

XC3000 Families Pin Assignments

Xilinx offers the six different array sizes in the XC3000 families in a variety of surface-mount and through-hole package types, with pin counts from 44 to 223.

Each chip is offered in several package types to accomodate the available pc board space and manufacturing technology. Most package types are also offered with different chips to accomodate design changes without the need for pc board changes.

Note that there is no perfect match between the number of bonding pads on the chip and the number of pins on a package. In some cases, the chip has more pads than there are pins on the package, as indicated by the information ("unused" pads) below the line in the following table. The IOBs of the unconnected pads can still be used as storage elements if the specified propagation delays and set-up times are acceptable.

In other cases, the chip has fewer pads than there are pins on the package; therefore, some package pins are not connected (n.c.), as shown above the line in the following table.

Number of Package Pins

Device	Pads	44	64	68	84	100	132	144	160	175	176	208	223
3020	74	—	—	6 u	10 n.c.	26 n.c.	—	—	—	—	—	—	—
3030	98	54 u	34 u	30 u	14 n.c.	2 n.c.	—	—	—	—	—	—	—
3042	118	—	—	—	34 u	18 u	14 n.c.	26 n.c.	—	—	—	—	—
3064	142	—	—	—	58 u	—	10 u	2 u	18 n.c.	—	—	—	—
3090	166	—	—	—	82 u	—	—	—	6 u	9 n.c	10 n.c.	42 n.c.	—
3195	198	—	—	—	114 u	—	—	—	—	9 n.c. 32 u	—	10 n.c.	25 n.c.

n.c. = Unconnected package pin
u = Unbonded device pad

X3457

Number of Available I/O Pins

		Number of Package Pins																
	Max I/O	44	64	68	84	100	120	132	144	156	160	164	175	176	191	196	208	223	240
XC3020/XC3120	64			58	64	64													
XC3030/XC3130	80	34	54	58	74	80													
XC3042/XC3142	96				74	82		96											
XC3064/XC3164	120				70			110			120								
XC3090/XC3190	144				70						138	142	144	144			144		
XC3195	176				70						138		144				176	176	

X3478

XC3000 Family 44-Pin PLCC Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

Pin No.	XC3030
1	**GND**
2	I/O
3	I/O
4	I/O
5	I/O
6	I/O
7	$\overline{\text{PWRDWN}}$
8	TCLKIN-I/O
9	I/O
10	I/O
11	I/O
12	**VCC**
13	I/O
14	I/O
15	I/O
16	M1-$\overline{\text{RDATA}}$
17	M0-RTRIG
18	M2-I/O
19	HDC-I/O
20	$\overline{\text{LDC}}$-I/O
21	I/O
22	$\overline{\text{INIT}}$-I/O

Pin No.	XC3030
23	**GND**
24	I/O
25	I/O
26	XTL2(IN)-I/O
27	$\overline{\text{RESET}}$
28	DONE-$\overline{\text{PGM}}$
29	I/O
30	XTL1(OUT)-BCLK-I/O
31	I/O
32	I/O
33	I/O
34	**VCC**
35	I/O
36	I/O
37	I/O
38	DIN-I/O
39	DOUT-I/O
40	CCLK
41	I/O
42	I/O
43	I/O
44	I/O

Peripheral mode and Master Parallel mode are not supported in the PC44 package

XC3030 Family 64-Pin Plastic VQFP Pinouts

XC3000, XC3000A, XC3000L and XC3100 families have idential pinouts

Pin No.	XC3030
1	A0-$\overline{\text{WS}}$-I/O
2	A1-CS2-I/O
3	A2-I/O
4	A3-I/O
5	A4-I/O
6	A14-I/O
7	A5-I/O
8	**GND**
9	A13-I/O
10	A6-I/O
11	A12-I/O
12	A7-I/O
13	A11-I/O
14	A8-I/O
15	A10-I/O
16	A9-I/O
17	$\overline{\text{PWRDN}}$
18	I/O
19	I/O
20	I/O
21	I/O
22	I/O
23	I/O
24	**VCC**
25	I/O
26	I/O
27	I/O
28	I/O
29	I/O
30	I/O
31	M1-$\overline{\text{RDATA}}$
32	M0-RTRIG

Pin No.	XC3030
33	M2-I/O
34	HDC-I/O
35	I/O
36	$\overline{\text{LDC}}$-I/O
37	I/O
38	I/O
39	I/O
40	$\overline{\text{INIT}}$-I/O
41	**GND**
42	I/O
43	I/O
44	I/O
45	I/O
46	I/O
47	XTAL2(IN)-I/O
48	$\overline{\text{RESET}}$
49	DONE-$\overline{\text{PG}}$
50	D7-I/O
51	XTAL1(OUT)-BCLKIN-I/O
52	D6-I/O
53	D5-I/O
54	$\overline{\text{CS0}}$-I/O
55	D4-I/O
56	**VCC**
57	D3-I/O
58	$\overline{\text{CS1}}$-I/O
59	D2-I/O
60	D1-I/O
61	RDY-$\overline{\text{BUSY}}$-$\overline{\text{RCLK}}$-I/O
62	D0-DIN-I/O
63	DOUT-I/O
64	CCLK

XC3000 Families 68-Pin PLCC, 84-Pin PLCC and PGA Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

68 PLCC		XC3020		
XC3030	XC3020	XC3030, XC3042	84 PLCC	84 PGA
10	10	PWRDN	12	B2
11	11	TCLKIN-I/O	13	C2
12	—	**I/O***	14	B1
13	12	I/O	15	C1
14	13	I/O	16	D2
—	—	I/O	17	D1
15	14	I/O	18	E3
16	15	I/O	19	E2
—	16	I/O	20	E1
17	17	I/O	21	F2
18	18	**VCC**	22	F3
19	19	I/O	23	G3
—	—	I/O	24	G1
20	20	I/O	25	G2
—	21	I/O	26	F1
21	22	I/O	27	H1
22	—	I/O	28	H2
23	23	I/O	29	J1
24	24	I/O	30	K1
25	25	M1-RDATA	31	J2
26	26	M0-RTRIG	32	L1
27	27	M2-I/O	33	K2
28	28	HDC-I/O	34	K3
29	29	I/O	35	L2
30	30	LDC-I/O	36	L3
—	31	I/O	37	K4
—		**I/O***	38	L4
31	32	I/O	39	J5
32	33	I/O	40	K5
33	—	**I/O***	41	L5
34	34	INIT-I/O	42	K6
35	35	**GND**	43	J6
36	36	I/O	44	J7
37	37	I/O	45	L7
38	38	I/O	46	K7
39	39	I/O	47	L6
—	40	I/O	48	L8
—	41	I/O	49	K8
40		**I/O***	50	L9
41		**I/O***	51	L10
42	42	I/O	52	K9
43	43	XTL2(IN)-I/O	53	L11

68 PLCC		XC3020		
XC3030	XC3020	XC3030, XC3042	84 PLCC	84 PGA
44		RESET	54	K10
45		DONE-PG	55	J10
46		D7-I/O	56	K11
47		XTL1(OUT)-BCLKIN-I/O	57	J11
48		D6-I/O	58	H10
—		I/O	59	H11
49		D5-I/O	60	F10
50		CS0-I/O	61	G10
51		D4-I/O	62	G11
—		I/O	63	G9
52		**VCC**	64	F9
53		D3-I/O	65	F11
54		CS1-I/O	66	E11
55		D2-I/O	67	E10
—		I/O	68	E9
—		**I/O***	69	D11
56		D1-I/O	70	D10
57		RDY/BUSY-RCLK-I/O	71	C11
58		D0-DIN-I/O	72	B11
59		DOUT-I/O	73	C10
60		CCLK	74	A11
61		A0-WS-I/O	75	B10
62		A1-CS2-I/O	76	B9
63		A2-I/O	77	A10
64		A3-I/O	78	A9
—		**I/O***	79	B8
—		**I/O***	80	A8
65		A15-I/O	81	B6
66		A4-I/O	82	B7
67		A14-I/O	83	A7
68		A5-I/O	84	C7
1		**GND**	1	C6
2		A13-I/O	2	A6
3		A6-I/O	3	A5
4		A12-I/O	4	B5
5		A7-I/O	5	C5
—		**I/O***	6	A4
—		**I/O***	7	B4
6		A11-I/O	8	A3
7		A8-I/O	9	A2
8		A10-I/O	10	B3
9		A9-I/O	11	A1

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

This table describes the pinouts of three different chips in three different packages. The pin-description column lists 84 of the 118 pads on the XC3042 (and 84 of the 98 pads on the XC3030) that are connected to the 84 package pins. Ten pads, indicated by an asterisk, do not exist on the XC3020, which has 74 pads; therefore the corresponding pins on the 84-pin packages have no connections to an XC3020. Six pads on the XC3020 and 16 pads on the XC3030, indicated by a dash (—) in the 68 PLCC column, have no connection to the 68 PLCC, but are connected to the 84-pin packages.

XC3064/XC3090/XC3195 84-Pin PLCC Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

PLCC Pin Number	XC3064, XC3090, XC3195
12	$\overline{\text{PWRDN}}$
13	TCLKIN-I/O
14	I/O
15	I/O
16	I/O
17	I/O
18	I/O
19	I/O
20	I/O
21	**GND***
22	VCC
23	I/O
24	I/O
25	I/O
26	I/O
27	I/O
28	I/O
29	I/O
30	I/O
31	M1-$\overline{\text{RDATA}}$
32	M0-RTRIG
33	M2-I/O
34	HDC-I/O
35	I/O
36	$\overline{\text{LDC}}$-I/O
37	I/O
38	I/O
39	I/O
40	I/O
41	INIT/I/O*
42	**VCC***
43	**GND**
44	I/O
45	I/O
46	I/O
47	I/O
48	I/O
49	I/O
50	I/O
51	I/O
52	I/O
53	XTL2(IN)-I/O

PLCC Pin Number	XC3064, XC3090, XC3195
54	$\overline{\text{RESET}}$
55	DONE-$\overline{\text{PG}}$
56	D7-I/O
57	XTL1(OUT)-BCLKIN-I/O
58	D6-I/O
59	I/O
60	D5-I/O
61	$\overline{\text{CS0}}$-I/O
62	D4-I/O
63	I/O
64	VCC
65	**GND***
66	D3-I/O*
67	$\overline{\text{CS1}}$-I/O*
68	D2-I/O*
69	I/O
70	D1-I/O
71	RDY/$\overline{\text{BUSY}}$-$\overline{\text{RCLK}}$-I/O
72	D0-DIN-I/O
73	DOUT-I/O
74	CCLK
75	A0-$\overline{\text{WS}}$-I/O
76	A1-CS2-I/O
77	A2-I/O
78	A3-I/O
79	I/O
80	I/O
81	A15-I/O
82	A4-I/O
83	A14-I/O
84	A5-I/O
1	**GND**
2	**VCC***
3	A13-I/O*
4	A6-I/O*
5	A12-I/O*
6	A7-I/O*
7	I/O
8	A11-I/O
9	A8-I/O
10	A10-I/O
11	A9-I/O

Unprogrammed IOBs have a default pull-up. This prevents an undfined pad level for unbonded or unused IOBs. Programmed ouptuts are default slew-rate limited.

*In the PC84 package, XC3064, XC3090 and XC3195 have additional V_{CC} and GND pins and thus a different pin definition than XC3020/XC3030/XC3042.

XC3000 Families 100-Pin QFP Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

Pin No.			XC3020 XC3030 XC3042
CQFP	PQFP	TQFP VQFP	
1	16	13	**GND**
2	17	14	A13-I/O
3	18	15	A6-I/O
4	19	16	A12-I/O
5	20	17	A7-I/O
6	21	18	**I/O***
7	22	19	**I/O***
8	23	20	A11-I/O
9	24	21	A8-I/O
10	25	22	A10-I/O
11	26	23	A9-I/O
12	27	24	**VCC***
13	28	25	**GND***
14	29	26	$\overline{\text{PWRDN}}$
15	30	27	TCLKIN-I/O
16	31	28	**I/O****
17	32	29	**I/O***
18	33	30	**I/O***
19	34	31	I/O
20	35	32	I/O
21	36	33	I/O
22	37	34	I/O
23	38	35	I/O
24	39	36	I/O
25	40	37	I/O
26	41	38	**VCC**
27	42	39	I/O
28	43	40	I/O
29	44	41	I/O
30	45	42	I/O
31	46	43	I/O
32	47	44	I/O
33	48	45	I/O
34	49	46	I/O

Pin No.			XC3020 XC3030 XC3042
CQFP	PQFP	TQFP VQFP	
35	50	47	**I/O***
36	51	48	**I/O***
37	52	49	M1-$\overline{\text{RD}}$
38	53	50	**GND***
39	54	51	MO-RT
40	55	52	**VCC***
41	56	53	M2-I/O
42	57	54	HDC-I/O
43	58	55	I/O
44	59	56	$\overline{\text{LDC}}$-I/O
45	60	57	**I/O***
46	61	58	I/O*
47	62	59	I/O
48	63	60	I/O
49	64	61	I/O
50	65	62	$\overline{\text{INIT}}$-I/O
51	66	63	**GND**
52	67	64	I/O
53	68	65	I/O
54	69	66	I/O
55	70	67	I/O
56	71	68	I/O
57	72	69	I/O
58	73	70	I/O
59	74	71	**I/O***
60	75	72	**I/O***
61	76	73	XTL2-I/O
62	77	74	**GND***
63	78	75	$\overline{\text{RESET}}$
64	79	76	**VCC***
65	80	77	DONE-$\overline{\text{PG}}$
66	81	78	D7-I/O
67	82	79	BCLKIN-XTL1-I/O
68	83	80	D6-I/O

Pin No.			XC3020 XC3030 XC3042
CQFP	PQFP	TQFP VQFP	
69	84	81	**I/O***
70	85	82	**I/O***
71	86	83	I/O
72	87	84	D5-I/O
73	88	85	$\overline{\text{CS0}}$-I/O
74	89	86	D4-I/O
75	90	87	I/O
76	91	88	**VCC**
77	92	89	D3-I/O
78	93	90	$\overline{\text{CS1}}$-I/O
79	94	91	D2-I/O
80	95	92	I/O
81	96	93	**I/O***
82	97	94	**I/O***
83	98	95	D1-I/O
84	99	96	$\overline{\text{RCLK}}$-$\overline{\text{BUSY}}$/RDY-I/O
85	100	97	DO-DIN-I/O
86	1	98	DOUT-I/O
87	2	99	CCLK
88	3	100	**VCC***
89	4	1	**GND***
90	5	2	AO-$\overline{\text{WS}}$-I/O
91	6	3	A1-CS2-I/O
92	7	4	**I/O****
93	8	5	A2-I/O
94	9	6	A3-I/O
95	10	7	**I/O***
96	11	8	**I/O***
97	12	9	A15-I/O
98	13	10	A4-I/O
99	14	11	A14-I/O
100	15	12	A5-I/O

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

* This table describes the pinouts of three different chips in three different packges. The pin-description column lists 100 of the 118 pads on the XC3042 that are connected to the 100 package pins. Two pads, indicated by double asterisks, do not exist on the XC3030, which has 98 pads; therefore the corresponding pins have no connections. Twenty-six pads, indicated by single or double asterisks, do not exist on the XC3020, which has 74 pads; therefore, the corresponding pins have no connections. (See table on page 2-133.)

XC3000 Families 132-Pin Ceramic and Plastic PGA Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

PGA Pin Number	XC3042 XC3064	PGA Pin Number	XC3042 XC3064	PGA Pin Number	XC3042 XC3064	PGA Pin Number	XC3042 XC3064
C4	**GND**	B13	M1-RD	P14	RESET	M3	DOUT-I/O
A1	PWRDN	C11	**GND**	M11	**VCC**	P1	CCLK
C3	I/O-TCLKIN	A14	M0-RT	N13	DONE-PG	M4	**VCC**
B2	I/O	D12	**VCC**	M12	D7-I/O	L3	**GND**
B3	I/O	C13	M2-I/O	P13	XTL1-I/O-BCLKIN	M2	A0-WS-I/O
A2	**I/O***	B14	HDC-I/O	N12	I/O	N1	A1-CS2-I/O
B4	I/O	C14	I/O	P12	I/O	M1	I/O
C5	I/O	E12	I/O	N11	D6-I/O	K3	I/O
A3	**I/O***	D13	I/O	M10	I/O	L2	A2-I/O
A4	I/O	D14	LDC-I/O	P11	**I/O***	L1	A3-I/O
B5	I/O	E13	**I/O***	N10	I/O	K2	I/O
C6	I/O	F12	I/O	P10	I/O	J3	I/O
A5	I/O	E14	I/O	M9	D5-I/O	K1	A15-I/O
B6	I/O	F13	I/O	N9	CS0-I/O	J2	A4-I/O
A6	I/O	F14	I/O	P9	**I/O***	J1	**I/O***
B7	I/O	G13	I/O	P8	**I/O***	H1	A14-I/O
C7	**GND**	G14	INIT-I/O	N8	D4-I/O	H2	A5-I/O
C8	**VCC**	G12	**VCC**	P7	I/O	H3	**GND**
A7	I/O	H12	**GND**	M8	**VCC**	G3	**VCC**
B8	I/O	H14	I/O	M7	**GND**	G2	A13-I/O
A8	I/O	H13	I/O	N7	D3-I/O	G1	A6-I/O
A9	I/O	J14	I/O	P6	CS1-I/O	F1	**I/O***
B9	I/O	J13	I/O	N6	**I/O***	F2	A12-I/O
C9	I/O	K14	I/O	P5	**I/O***	E1	A7-I/O
A10	I/O	J12	I/O	M6	D2-I/O	F3	I/O
B10	I/O	K13	I/O	N5	I/O	E2	I/O
A11	**I/O***	L14	**I/O***	P4	I/O	D1	A11-I/O
C10	I/O	L13	I/O	P3	I/O	D2	A8-I/O
B11	I/O	K12	I/O	M5	D1-I/O	E3	I/O
A12	**I/O***	M14	I/O	N4	RCLK-BUSY/RDY-I/O	C1	I/O
B12	I/O	N14	I/O	P2	I/O	B1	A10-I/O
A13	**I/O***	M13	XTL2(IN)-I/O	N3	I/O	C2	A9-I/O
C12	I/O	L12	**GND**	N2	D0-DIN-I/O	D3	**VCC**

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

* Indicates unconnected package pins (14) for the XC3042.

XC3030/XC3064 Families 144-Pin Plastic TQFP Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

Pin Number	XC3042 XC3064
1	PWRDN
2	I/O-TCLKIN
3	I/O*
4	I/O
5	I/O
6	I/O*
7	I/O
8	I/O
9	I/O*
10	I/O
11	I/O
12	I/O
13	I/O
14	I/O
15	I/O*
16	I/O
17	I/O
18	**GND**
19	**VCC**
20	I/O
21	I/O
22	I/O
23	I/O
24	I/O
25	I/O
26	I/O
27	I/O
28	I/O*
29	I/O
30	I/O
31	I/O*
32	I/O*
33	I/O
34	I/O*
35	I/O
36	M1-RD
37	**GND**
38	MO-RT
39	**VCC**
40	M2-I/O
41	HDC-I/O
42	I/O
43	I/O
44	I/O
45	LDC-I/O
46	I/O*
47	I/O
48	I/O

Pin Number	XC3042 XC3064
49	I/O
50	I/O*
51	I/O
52	I/O
53	INIT-I/O
54	**VCC**
55	**GND**
56	I/O
57	I/O
58	I/O
59	I/O
60	I/O
61	I/O
62	I/O
63	I/O*
64	I/O*
65	I/O
66	I/O
67	I/O
68	I/O
69	XTL2(IN)-I/O
70	**GND**
71	RESET
72	**VCC**
73	DONE-PG
74	D7-I/O
75	XTL1(OUT)-BCLKIN-I/O
76	I/O
77	I/O
78	D6-I/O
79	I/O
80	I/O*
81	I/O
82	I/O
83	I/O*
84	D5-I/O
85	CS0-I/O
86	I/O*
87	I/O*
88	D4-I/O
89	I/O
90	**VCC**
91	**GND**
92	D3-I/O
93	CS1-I/O
94	I/O*
95	I/O*
96	D2-I/O

Pin Number	XC3042 XC3064
97	I/O
98	I/O
99	I/O*
100	I/O
101	I/O*
102	D1-I/O
103	RCLK-BUSY/RDY-I/O
104	I/O
105	I/O
106	D0-DIN-I/O
107	DOUT-I/O
108	CCLK
109	**VCC**
110	**GND**
111	A0-WSI/O
112	A1-CS2-I/O
113	I/O
114	I/O
115	A2-I/O
116	A3-I/O
117	I/O
118	I/O
119	A15-I/O
120	A4-I/O
121	I/O*
122	I/O*
123	A14-I/O
124	A5-I/O
125	–
126	**GND**
127	**VCC**
128	A13-I/O
129	A6-I/O
130	I/O*
131	–
132	I/O*
133	A12-I/O
134	A7-I/O
135	I/O
136	I/O
137	A11-I/O
138	A8-I/O
139	I/O
140	I/O
141	A10-I/O
142	A9-I/O
143	**VCC**
144	**GND**

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

* Indicates unconnected package pins (24) for the XC3042.

XC3000 Families160-Pin PQFP Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

PQFP Pin Number	XC3064, XC3090, XC3195
1	I/O*
2	I/O*
3	I/O*
4	I/O
5	I/O
6	I/O
7	I/O
8	I/O
9	I/O
10	I/O
11	I/O
12	I/O
13	I/O
14	I/O
15	I/O
16	I/O
17	I/O
18	I/O
19	**GND**
20	**VCC**
21	I/O*
22	I/O
23	I/O
24	I/O
25	I/O
26	I/O
27	I/O
28	I/O
29	I/O
30	I/O
31	I/O
32	I/O
33	I/O
34	I/O
35	I/O
36	I/O
37	I/O
38	I/O*
39	I/O*
40	M1-$\overline{\text{RDATA}}$

PQFP Pin Number	XC3064, XC3090, XC3195
41	**GND**
42	M0–RTRIG
43	**VCC**
44	M2-I/O
45	HDC-I/O
46	I/O
47	I/O
48	I/O
49	$\overline{\text{LDC}}$-I/O
50	I/O*
51	I/O*
52	I/O
53	I/O
54	I/O
55	I/O
56	I/O
57	I/O
58	I/O
59	$\overline{\text{INIT}}$-I/O
60	**VCC**
61	**GND**
62	I/O
63	I/O
64	I/O
65	I/O
66	I/O
67	I/O
68	I/O
69	I/O
70	I/O
71	I/O
72	I/O
73	I/O
74	I/O
75	I/O*
76	XTL2-I/O
77	**GND**
78	$\overline{\text{RESET}}$
79	**VCC**
80	DONE/$\overline{\text{PG}}$

PQFP Pin Number	XC3064, XC3090, XC3195
81	D7-I/O
82	XTL1-I/O-BCLKIN
83	I/O*
84	I/O
85	I/O
86	D6-I/O
87	I/O
88	I/O
89	I/O
90	I/O
91	I/O
92	D5-I/O
93	$\overline{\text{CS0}}$-I/O
94	I/O*
95	I/O*
96	I/O
97	I/O
98	D4-I/O
99	I/O
100	**VCC**
101	**GND**
102	D3-I/O
103	$\overline{\text{CS1}}$-I/O
104	I/O
105	I/O
106	I/O*
107	I/O*
108	D2-I/O
109	I/O
110	I/O
111	I/O
112	I/O
113	I/O
114	D1-I/O
115	RDY-$\overline{\text{BSY}}$/$\overline{\text{RCLK}}$-I/O
116	I/O
117	I/O
118	I/O*
119	D0-DIN-I/O
120	DOUT-I/O

PQFP Pin Number	XC3064, XC3090, XC3195
121	CCLK
122	**VCC**
123	**GND**
124	A0-$\overline{\text{WS}}$-I/O
125	A1-CS2-I/O
126	I/O
127	I/O
128	A2-I/O
129	A3-I/O
130	I/O
131	I/O
132	A15-I/O
133	A4-I/O
134	I/O
135	I/O
136	A14-I/O
137	A5-I/O
138	I/O*
139	**GND**
140	**VCC**
141	A13-I/O
142	A6-I/O
143	I/O*
144	I/O*
145	I/O
146	I/O
147	A12-I/O
148	A7-I/O
149	I/O
150	I/O
151	A11-I/O
152	A8-I/O
153	I/O
154	I/O
155	A10-I/O
156	A9-I/O
157	**VCC**
158	**GND**
159	$\overline{\text{PWRDWN}}$
160	TCLKIN-I/O

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed IOBs are default slew-rate limited.

*Indicates unconnected package pins (18) for the XC3064.

XC3000 Families 175-Pin Ceramic and Plastic PGA Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

PGA Pin Number	XC3090, XC3195
B2	PWRDN
D4	TCLKIN-I/O
B3	I/O
C4	I/O
B4	I/O
A4	I/O
D5	I/O
C5	I/O
B5	I/O
A5	I/O
C6	I/O
D6	I/O
B6	I/O
A6	I/O
B7	I/O
C7	I/O
D7	I/O
A7	I/O
A8	I/O
B8	I/O
C8	I/O
D8	**GND**
D9	**VCC**
C9	I/O
B9	I/O
A9	I/O
A10	I/O
D10	I/O
C10	I/O
B10	I/O
A11	I/O
B11	I/O
D11	I/O
C11	I/O
A12	I/O
B12	I/O
C12	I/O
D12	I/O
A13	I/O
B13	I/O
C13	I/O
A14	I/O

PGA Pin Number	XC3090, XC3195
D13	I/O
B14	M1-RDATA
C14	**GND**
B15	M0-RTRIG
D14	**VCC**
C15	M2-I/O
E14	HDC-I/O
B16	I/O
D15	I/O
C16	I/O
D16	LDC-I/O
F14	I/O
E15	I/O
E16	I/O
F15	I/O
F16	I/O
G14	I/O
G15	I/O
G16	I/O
H16	I/O
H15	INIT-I/O
H14	**VCC**
J14	**GND**
J15	I/O
J16	I/O
K16	I/O
K15	I/O
K14	I/O
L16	I/O
L15	I/O
M16	I/O
M15	I/O
L14	I/O
N16	I/O
P16	I/O
N15	I/O
R16	I/O
M14	I/O
P15	XTL2(IN)-I/O
N14	**GND**
R15	RESET
P14	**VCC**

PGA Pin Number	XC3090, XC3195
R14	DONE-PG
N13	D7-I/O
T14	XTL1(OUT)-BCLKIN-I/O
P13	I/O
R13	I/O
T13	I/O
N12	I/O
P12	D6-I/O
R12	I/O
T12	I/O
P11	I/O
N11	I/O
R11	I/O
T11	D5-I/O
R10	CS0-I/O
P10	I/O
N10	I/O
T10	I/O
T9	I/O
R9	D4-I/O
P9	I/O
N9	**VCC**
N8	**GND**
P8	D3-I/O
R8	CS1-I/O
T8	I/O
T7	I/O
N7	I/O
P7	I/O
R7	D2-I/O
T6	I/O
R6	I/O
N6	I/O
P6	I/O
T5	I/O
R5	D1-I/O
P5	RDY/BUSY-RCLK-I/O
N5	I/O
T4	I/O
R4	I/O
P4	I/O

PGA Pin Number	XC3090, XC3195
R3	D0-DIN-I/O
N4	DOUT-I/O
R2	CCLK
P3	**VCC**
N3	**GND**
P2	A0-WS-I/O
M3	A1-CS2-I/O
R1	I/O
N2	I/O
P1	A2-I/O
N1	A3-I/O
L3	I/O
M2	I/O
M1	A15-I/O
L2	A4-I/O
L1	I/O
K3	I/O
K2	A14-I/O
K1	A5-I/O
J1	I/O
J2	I/O
J3	**GND**
H3	**VCC**
H2	A13-I/O
H1	A6-I/O
G1	I/O
G2	I/O
G3	I/O
F1	I/O
F2	A12-I/O
E1	A7-I/O
E2	I/O
F3	I/O
D1	A11-I/O
C1	A8-I/O
D2	I/O
B1	I/O
E3	A10-I/O
C2	A9-I/O
D3	**VCC**
C3	**GND**

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

Pins A2, A3, A15, A16, T1, T2, T3, T15 and T16 are not connected. Pin A1 does not exist.

XC3090 176-Pin TQFP Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

Pin Number	XC3090	Pin Number	XC3090	Pin Number	XC3090	Pin Number	XC3090
1	$\overline{\text{PWRDWN}}$	45	M1-$\overline{\text{RDATA}}$	89	DONE-$\overline{\text{PG}}$	133	**VSS**
2	TCLKIN-I/O	46	**GND**	90	D7-I/O	134	**GND**
3	I/O	47	M0-RTRIG	91	XTAL1(OUT)-BCLKIN-I/O	135	A0-$\overline{\text{WS}}$-I/O
4	I/O	48	**VCC**	92	I/O	136	A1-CS2-I/O
5	I/O	49	M2-I/O	93	I/O	137	–
6	I/O	50	HDC-I/O	94	I/O	138	I/O
7	I/O	51	I/O	95	I/O	139	I/O
8	I/O	52	I/O	96	D6-I/O	140	A2-I/O
9	I/O	53	I/O	97	I/O	141	A3-I/O
10	I/O	54	$\overline{\text{LDC}}$-I/O	98	I/O	142	–
11	I/O	55	–	99	I/O	143	–
12	I/O	56	I/O	100	I/O	144	I/O
13	I/O	57	I/O	101	I/O	145	I/O
14	I/O	58	I/O	102	D5-I/O	146	A15-I/O
15	I/O	59	I/O	103	$\overline{\text{CS0}}$-I/O	147	A4-I/O
16	I/O	60	I/O	104	I/O	148	I/O
17	I/O	61	I/O	105	I/O	149	I/O
18	I/O	62	I/O	106	I/O	150	A14-I/O
19	I/O	63	I/O	107	I/O	151	A5-I/O
20	I/O	64	I/O	108	D4-I/O	152	I/O
21	I/O	65	$\overline{\text{INIT}}$-I/O	109	I/O	153	I/O
22	**GND**	66	**VCC**	110	**VCC**	154	**GND**
23	**VCC**	67	**GND**	111	**GND**	155	**VCC**
24	I/O	68	I/O	112	D3-I/O	156	A13-I/O
25	I/O	69	I/O	113	$\overline{\text{CS1}}$-I/O	157	A6-I/O
26	I/O	70	I/O	114	I/O	158	I/O
27	I/O	71	I/O	115	I/O	159	I/O
28	I/O	72	I/O	116	I/O	160	–
29	I/O	73	I/O	117	I/O	161	–
30	I/O	74	I/O	118	D2-I/O	162	I/O
31	I/O	75	I/O	119	I/O	163	I/O
32	I/O	76	I/O	120	I/O	164	A12-I/O
33	I/O	77	I/O	121	I/O	165	A7-I/O
34	I/O	78	I/O	122	I/O	166	I/O
35	I/O	79	I/O	123	I/O	167	I/O
36	I/O	80	I/O	124	D1-I/O	168	–
37	I/O	81	I/O	125	RDY/$\overline{\text{BUSY}}$-RCLK-I/O	169	A11-I/O
38	I/O	82	–	126	I/O	170	A8-I/O
39	I/O	83	–	127	I/O	171	I/O
40	I/O	84	I/O	128	I/O	172	I/O
41	I/O	85	XTAL2(IN)-I/O	129	I/O	173	A10-I/O
42	I/O	86	**GND**	130	D0-DIN-I/O	174	A9-I/O
43	I/O	87	$\overline{\text{RESET}}$	131	DOUT-I/O	175	**VCC**
44	–	88	**VCC**	132	CCLK	176	**GND**

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

XC3090 208-Pin PQFP Pinouts

XC3000, XC3000A, XC3000L, XC3100 and XC3100A families have identical pinouts

Pin Number	XC3090	Pin Number	XC3090	Pin Number	XC3090	Pin Number	XC3090
1	–	53	–	105	–	157	–
2	**GND**	54	–	106	**VCC**	158	–
3	PWRDWN	55	**VCC**	107	D/P	159	–
4	TCLKIN-I/O	56	M2-I/O	108	–	160	**GND**
5	I/O	57	HDC-I/O	109	D7-I/O	161	WS-A0-I/O
6	I/O	58	I/O	110	XTL1-BCLKIN-I/O	162	CS2-A1-I/O
7	I/O	59	I/O	111	I/O	163	I/O
8	I/O	60	I/O	112	I/O	164	I/O
9	I/O	61	LDC-I/O	113	I/O	165	A2-I/O
10	I/O	62	I/O	114	I/O	166	A3-I/O
11	I/O	63	I/O	115	D6-I/O	167	I/O
12	I/O	64	–	116	I/O	168	I/O
13	I/O	65	–	117	I/O	169	–
14	I/O	66	–	118	I/O	170	–
15	–	67	–	119	–	171	–
16	I/O	68	I/O	120	I/O	172	A15-I/O
17	I/O	69	I/O	121	I/O	173	A4-I/O
18	I/O	70	I/O	122	D5-I/O	174	I/O
19	I/O	71	I/O	123	CS0-I/O	175	I/O
20	I/O	72	–	124	I/O	176	–
21	I/O	73	–	125	I/O	177	–
22	I/O	74	I/O	126	I/O	178	A14-I/O
23	I/O	75	I/O	127	I/O	179	A5-I/O
24	I/O	76	I/O	128	D4-I/O	180	I/O
25	**GND**	77	INIT-I/O	129	I/O	181	I/O
26	**VCC**	78	**VCC**	130	**VCC**	182	**GND**
27	I/O	79	**GND**	131	**GND**	183	**VCC**
28	I/O	80	I/O	132	D3-I/O	184	A13-I/O
29	I/O	81	I/O	133	CS1-I/O	185	A6-I/O
30	I/O	82	I/O	134	I/O	186	I/O
31	I/O	83	–	135	I/O	187	I/O
32	I/O	84	–	136	I/O	188	–
33	I/O	85	I/O	137	I/O	189	–
34	I/O	86	I/O	138	D2-I/O	190	I/O
35	I/O	87	I/O	139	I/O	191	I/O
36	I/O	88	I/O	140	I/O	192	A12-I/O
37	–	89	I/O	141	I/O	193	A7-I/O
38	I/O	90	–	142	–	194	–
39	I/O	91	–	143	I/O	195	–
40	I/O	92	–	144	I/O	196	–
41	I/O	93	I/O	145	D1-I/O	197	I/O
42	I/O	94	I/O	146	BUSY/RDY-RCLK-I/O	198	I/O
43	I/O	95	I/O	147	I/O	199	A11-I/O
44	I/O	96	I/O	148	I/O	200	A8-I/O
45	I/O	97	I/O	149	I/O	201	I/O
46	I/O	98	I/O	150	I/O	202	I/O
47	I/O	99	I/O	151	DIN-D0-I/O	203	A10-I/O
48	M1-RDATA	100	XTL2-I/O	152	DOUT-I/O	204	A9-I/O
49	**GND**	101	**GND**	153	CCLK	205	**VCC**
50	M0-RTRIG	102	RESET	154	**VCC**	206	–
51	–	103	–	155	–	207	–
52	–	104	–	156	–	208	–

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

*In PQ208, XC3090 and XC3195 have different pinouts.

XC3195 PQ208 and PG223 Pinouts

Pin Description	PG223	PQ208 *
A9-I/O	B1	206
A10-I/O	E3	205
I/O	E4	204
I/O	C2	203
I/O	C1	202
I/O	D2	201
A8-I/O	E2	200
A11-I/O	F4	199
I/O	F3	198
I/O	D1	197
I/O	F2	196
I/O	G2	194
A7-I/O	G4	193
A12-I/O	G1	192
I/O	H2	191
I/O	H3	190
I/O	H1	189
I/O	H4	188
I/O	J3	187
I/O	J2	186
A6-I/O	J1	185
A13-I/O	K3	184
VCC	J4	183
GND	K4	182
I/O	K2	181
I/O	K1	180
A5-I/O	L2	179
A14-I/O	L4	178
I/O	L3	177
I/O	L1	176
I/O	M1	175
I/O	M2	174
A4-I/O	M4	173
A15-I/O	N2	172
I/O	N3	171
I/O	P2	169
I/O	R1	168
I/O	N4	167
A3-I/O	T1	166
A2-I/O	R2	165
I/O	P3	164
I/O	T2	163
I/O	P4	162
I/O	U1	161
A1-CS2-I/O	V1	160
A0-WS-I/O	T3	159
GND	R3	158
VCC	R4	157
CCLK	U2	156
DOUT-I/O	V2	155

Pin Description	PG223	PQ208 *
D0-DIN-I/O	U3	154
I/O	V3	153
I/O	R5	152
I/O	T4	151
I/O	V4	150
RDY/BUSY-RCLK-I/O	U4	149
D1-I/O	U5	148
I/O	R6	147
I/O	T5	146
I/O	U6	145
I/O	T6	144
I/O	V7	141
I/O	R7	140
I/O	U7	139
D2-I/O	V8	138
I/O	U8	137
I/O	T8	136
I/O	R8	135
I/O	V9	134
$\overline{CS1}$-I/O	U9	133
D3-I/O	T9	132
GND	R9	131
VCC	R10	130
I/O	T10	129
D4-I/O	U10	128
I/O	V10	127
I/O	R11	126
I/O	T11	125
I/O	U11	124
$\overline{CS0}$-I/O	V11	123
D5-I/O	U12	122
I/O	R12	121
I/O	V12	120
I/O	T13	119
I/O	U13	118
I/O	T14	117
I/O	R13	116
I/O	U14	115
D6-I/O	U15	114
I/O	V15	113
I/O	T15	112
I/O	R14	111
I/O	V16	110
XTL1(OUT)BCLKIN-I/O	U16	109
D7-I/O	T16	108
D/P	V17	107
VCC	R15	106
$\overline{RESET}$	U17	105
GND	R16	104
XTL2(IN)-I/O	V18	103

Pin Description	PG223	PQ208 *
I/O	U18	102
I/O	P15	101
I/O	T17	100
I/O	T18	99
I/O	P16	98
I/O	R17	97
I/O	N15	96
I/O	R18	95
I/O	P17	94
I/O	N17	93
I/O	N16	92
I/O	M15	89
I/O	M18	88
I/O	M17	87
I/O	L18	86
I/O	L17	85
I/O	L15	84
I/O	L16	83
I/O	K18	82
I/O	K17	81
I/O	K16	80
GND	K15	79
VCC	J15	78
$\overline{INIT}$	J16	77
I/O	J17	76
I/O	J18	75
I/O	H16	74
I/O	H15	73
I/O	H17	72
I/O	H18	71
I/O	G17	70
I/O	G18	69
I/O	G15	68
I/O	F16	67
I/O	F17	66
I/O	E17	63
I/O	C18	62
I/O	F15	61
I/O	D17	60
$\overline{LDC}$-I/O	E16	59
I/O	C17	58
I/O	B18	57
I/O	E15	56
HDC-I/O	A18	55
M2-I/O	A17	54
VCC	D16	53
M0-RTIG	B17	52
GND	D15	51
M1/$\overline{RDATA}$	C16	50

Pin Description	PG223	PQ208 *
I/O	B16	49
I/O	A16	48
I/O	D14	47
I/O	C15	46
I/O	B15	45
I/O	A15	44
I/O	C14	43
I/O	D13	42
I/O	B14	41
I/O	C13	40
I/O	B13	39
I/O	B12	38
I/O	D12	37
I/O	A12	36
I/O	B11	35
I/O	C11	34
I/O	A11	33
I/O	D11	32
I/O	A10	31
I/O	B10	30
I/O	C10	29
I/O	C9	28
VCC	D10	27
GND	D9	26
I/O	B9	25
I/O	A9	24
I/O	C8	23
I/O	D8	22
I/O	B8	21
I/O	A8	20
I/O	B7	19
I/O	A7	18
I/O	D7	17
I/O	B6	14
I/O	C6	13
I/O	B5	12
I/O	A4	11
I/O	D6	10
I/O	C5	9
I/O	B4	8
I/O	B3	7
I/O	C4	6
I/O	D5	5
I/O	C3	4
I/O	A3	3
TCLKIN-I/O	A2	2
$\overline{PWRDN}$	B2	1
GND	D4	208
VCC	D3	207

Unprogrammed IOBs have a default pull-up. This prevents an undefined pad level for unbonded or unused IOBs. Programmed outputs are default slew-rate limited.

In the PQ208 package, pins 15, 16, 64, 65, 90, 91, 142, 143, 170 and 195 are not connected.

*In PQ208, XC3090 and XC3195 have different pinouts.

XC3000 Component Availability

PINS		44	64	68	84		100				132		144	160	164	175		176	208	223
TYPE		PLAST. PLCC	PLAST. VQFP	PLAST. PLCC	PLAST. PLCC	CERAM PGA	PLAST. PQFP	PLAST. TQFP	PLAST. VQFP	TOP-BRAZED CQFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	TOP-BRAZED CQFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	CERAM. PGA
CODE		PC44	VQ64	PC68	PC84	PG84	PQ100	TQ100	VQ100	CB100	PP132	PG132	TQ144	PQ160	CB164	PP175	PG175	TQ176	PQ208	PG223
XC3020	-50					M B				M B										
	-70			C I	C I	C I M B	C I			C M B										
	-100			C I	C I	C I M B	C I			C M B										
	-125			C	C	C	C													
XC3030	-50					M														
	-70	C I		C I	C I	C I M	C I	C												
	-100	C I		C I	C I	C I M	C I	C												
	-125	C		C	C	C	C	C												
XC3042	-50					M B				M B		M B								
	-70				C I	C I M B	C I	C		C M B	C	C I M B								
	-100				C I	C I M B	C I	C		C M B	C	C I M B								
	-125				C	C	C	C			C	C								
XC3064	-50											M								
	-70				C I						C I	C I M		C I						
	-100				C I						C I	C I M		C I						
	-125				C						C	C		C						
XC3090	-50														M B		M B			
	-70				C I									C I	C M B	C I	C I M B		C I	
	-100				C I									C I	C M B	C I	C I M B		C I	
	-125				C									C		C	C		C	
XC3020A	-7			C I	C I	C I	C I													
	-6			C	C	C	C													
XC3030A	-7	C I	C I	C I	C I	C I	C I		C I											
	-6	C	C	C	C	C	C		C											
XC3042A	-7				C I	C I	C I		C I		C I	C I	C I							
	-6				C	C	C		C		C	C	C							
XC3064A	-7				C I						C I	C I	C I	C I						
	-6				C						C	C	C	C						
XC3090A	-7				C I									C I		C I	C I	C I	C I	
	-6				C									C		C	C	C	C	
XC3020L					C															
XC3030L			C		C				C											
XC3042L					C				C				C							
XC3064L					C								C							
XC3090L					C													C		
XC3120 XC3120A	-5			C I	C I	C I	C I (M B)			(M B)										
	-4			C I	C I	C I	C I													
	-3			C	C	C	C													
XC3130 XC3130A	-5	C I		C I	C I	C I	C I	C												
	-4	C I		C I	C I	C I	C I	C												
	-3	C		C	C	C	C	C												
XC3142 XC3142A	-5				C I	C I	C I (M B)	C		(M B)	C	C I (M B)	C I							
	-4				C I	C I	C I	C			C	C I	C I							
	-3				C	C	C	C			C	C	C							
XC3164 XC3164A	-5				C I						C I	C I		C I						
	-4				C I						C I	C I		C I						
	-3				C						C	C		C						
XC3190 XC3190A	-5				C I									C I	(M B)	C I	C I (M B)		C I	
	-4				C I									C I		C I	C I		C I	
	-3				C									C		C	C		C	
XC3195 XC3195A	-5				C I									C I		C I	C I (M B)		C I	C I (M B)
	-4				C I									C I		C I	C I		C I	C I
	-3				C									C		C	C		C	C

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C B = MIL-STD-883C Class B
Parentheses indicate future product plans

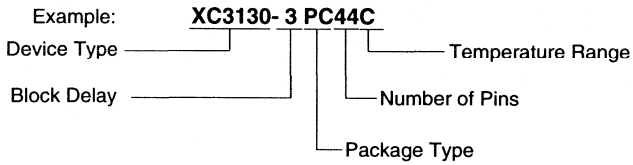
For a detailed description of the device architecture, see pages 2-105 through 2-123.

For a detailed description of the configuration modes and their timing, see pages 2-124 through 2-132.

For detailed lists of package pin-outs, see pages 2-140 through 2-150.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



XC3000, XC3000A, XC3000L, XC3100, XC3100A

The features of the original **XC3000** family are described on the preceding pages.

XC3100 is functionally identical with XC3000, but offers substantially faster performance. There is also an additional high-end family member, the XC3195.

XC3000L uses a 3.3 V supply voltage and has lower power-down current.

The **XC3000A, XC3000L** and **XC3100A** families all offer identical enhanced functionality. They are thus supersets of the XC3000 and XC3100 families:

Additional routing resources provide improved performance and higher density. There is now a direct connection from each CLB output to the data input of its nearest TBUF. This speeds up the path and preserves general routing resources that can be used for other purposes.

The **CLB clock enable and the TBUF output enable** are now driven by two different vertical Longlines. In the XC3000/3100 devices, the CLB clock enable signal and the adjacent TBUF output enable signal can both be driven only from the same vertical Longline. That makes these two functions mutually exclusive, and thus creates placement constraints. Using separate Longlines for these two functions leads to improved density and performance, especially in bus-oriented applications.

Bitstream error checking protects against erroneous configuration.

Each Xilinx FPGA bitstream consists of a 40-bit preamble, followed by a device-specific number of data frames. The number of bits per frame is also device-specific; however, each frame ends with three stop bits (111) followed by a start bit for the next frame (0).

All devices in all XC3000 families start reading in a new frame when they find the first 0 after the end of the previous frame. XC3000/XC3100 devices do not check for the correct stop bits, but XC3000A/XC3100A and XC3000L devices check that the last three bits of any frame are actually 111.

Under normal circumstances, all these FPGAs behave the same way; however, if the bitstream is corrupted, an XC3000/XC3100 device will always start a new frame as soon as it finds the first 0 after the end of the previous frame, even if the data is completely wrong or out-of-sync. Given sufficient zeros in the data stream, the device will also go Done, but with incorrect configuration and the possibility of internal contention.

An XC3000A/XC3100A/XC3000L device starts any new frame only if the three preceding bits are all ones. If this check fails, it pulls $\overline{\text{INIT}}$ Low and stops the internal configuration, although the Master CCLK keeps running. The user must then start a new configuration by applying a >6 µs Low level on $\overline{\text{RESET}}$.

This simple check does not protect against random bit errors, but it offers almost 100 percent protection against erroneous configuration files, defective configuration data sources, synchronization errors between configuration source and FPGA, or PC-board level defects, such as broken lines or solder-bridges.

A separate modification slows down the $\overline{\text{RESET}}$ input before configuration by using a two-stage shift register driven from the internal clock. It tolerates submicrosecond High spikes on $\overline{\text{RESET}}$ before configuration, and simplifies the Shorter Power-on Delay application described on page 9-xx. The XC3000 master can be connected like an XC4000 master, but with its $\overline{\text{RESET}}$ input used instead of $\overline{\text{INIT}}$. (On XC3000, $\overline{\text{INIT}}$ is output only).

Soft start-up. After configuration, the outputs of all LCA device in a daisy-chain become active simultaneously, as a result of the same CCLK edge. In the original XC3000/3100 devices, each output becomes active in either fast or slew-rate limited mode, depending on the way it is configured. This can lead to large ground-bounce signals. In the new XC3000A/XC3000L/XC31000A devices, all outputs become active first in slew-rate limited mode, reducing the ground bounce. After this soft start-up, each individual output slew rate is again controlled by the respective configuration bit.

XC3000
Logic Cell Array Family

Product Specification

Features

- Industry-leading FPGA family with five device types
  - Logic densities from 1,000 to 6,000 gates
  - Up to 144 user-definable I/Os
- Guaranteed 70- to 125-MHz toggle rates, 9 to 5.5 ns logic delays
- Advanced CMOS static memory technology
  - Low quiescent and active power consumption
- XC3000-specific features
  - Ultra-low current option in Power-Down mode
  - 4-mA output sink and source current
  - Broad range of package options includes plastic and ceramic quad flat packs, plastic leaded chip carriers and pin grid arrays
  - 100% bitstream compatible with the XC3100 family
  - Commercial, industrial, military, "high rel", and MIL-STD-883 Class B grade devices
  - Easy migration to XC3300 series of HardWire mask-programmed devices for high-volume production

Description

XC3000 is the original family of devices in the XC3000 class of Field Programmable Gate Array (FPGA) architectures. The XC3000 family has a proven track record in addressing a wide range of design applications, including general logic replacement and sub-systems integration. For a thorough description of the XC3000 architecture see the preceding pages of this data book.

The XC3000 Family covers a range of nominal device densities from 2,000 to 9,000 gates, practically achievable densities from 1,000 to 6,000 gates. Device speeds, described in terms of maximum guaranteed toggle frequencies, range from 70 to 125 MHz. The performance of a completed design depends upon placement and routing implementation, so, like with any gate array, the final verification of device utilization and performance can only be known after the design has been placed and routed.

Device	CLBs	Array	User I/Os Max	Flip-Flops	Horizontal Longlines	Configuration Data Bits
XC3020	64	8 x 8	64	256	16	14,779
XC3030	100	10 x 10	80	360	20	22,176
XC3042	144	12 x 12	96	480	24	30,784
XC3064	224	16 x 14	120	688	28	46,064
XC3090	320	16 x 20	144	928	40	64,160

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to VCC +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C
T_J	Junction temperature plastic	+125	°C
	Junction temperature ceramic	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description	Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial 0°C to +70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial –40°C to +85°C	4.5	5.5	V
V_{IHT}	High-level input voltage — TTL configuration	2.0	V_{CC}	V
V_{ILT}	Low-level input voltage — TTL configuration	0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration	70%	100%	V_{CC}
V_{ILC}	Low-level input voltage — CMOS configuration	0	20%	V_{CC}
T_{IN}	Input signal transition time		250	ns

DC Characteristics Over Operating Conditions

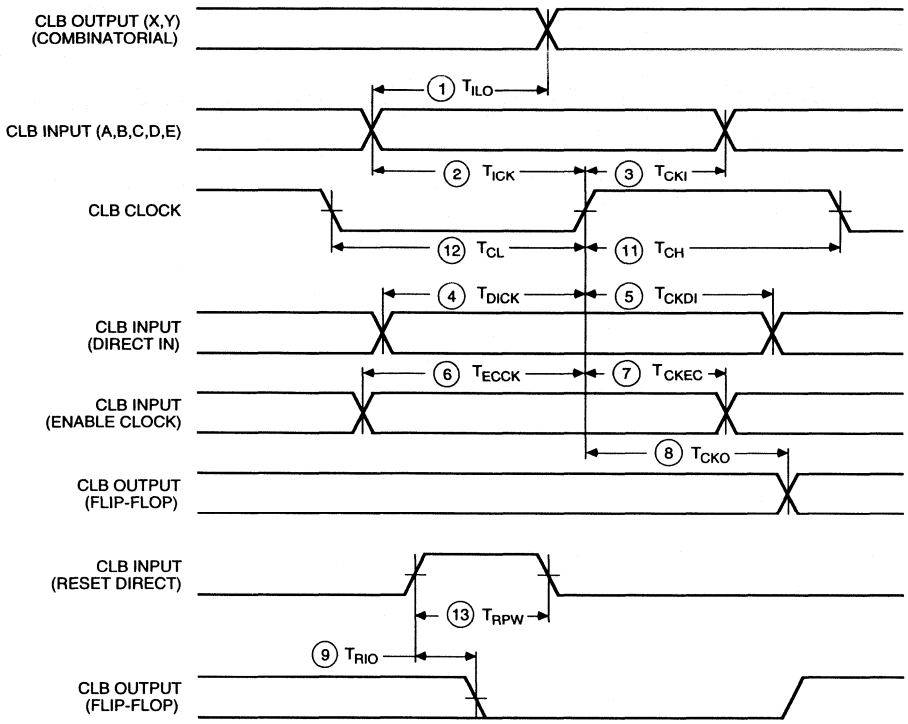
Symbol	Description		Min	Max	Units
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 mA, V_{CC} min)	Commercial	3.86		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} max)	Commercial		0.40	V
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 mA, V_{CC} min)	Industrial	3.76		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} max)	Industrial		0.40	V
V_{CCPD}	Power-down supply voltage ($\overline{PWRDWN}$ must be Low)		2.30		V
I_{CCPD}	Power-down supply current ($V_{CC(MAX)}$ @ T_{MAX})[1]	XC3020		50	µA
		XC3030		80	µA
		XC3042		120	µA
		XC3064		170	µA
		XC3090		250	µA
I_{CCO}	Quiescent LCA supply current in addition to I_{CCPD}[2] Chip thresholds programmed as CMOS levels			500	µA
	Chip thresholds programmed as TTL levels			10	mA
I_{IL}	Input Leakage Current		–10	+10	µA
C_{IN}	Input capacitance, all packages except PGA175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			 10 15	 pF pF
	Input capacitance, PGA 175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			 15 20	 pF pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0 V (sample tested)		0.02	0.17	mA
I_{RLL}	Horizontal Longline pull-up (when selected) @ logic Low			3.4	mA

Note: 1. Devices with much lower I_{CCPD} tested and guaranteed at V_{CC} = 3.2 V, T = 25°C can be ordered with a Special Product Code.

XC3020 SPC0107: I_{CCPD} = 1 µA
XC3030 SPC0107: I_{CCPD} = 2 µA
XC3042 SPC0107: I_{CCPD} = 3 µA
XC3064 SPC0107: I_{CCPD}= 4 µA
XC3090 SPC0107: I_{CCPD}= 5 µA

2. With no output current loads, no active input or Longline pull-up resistors, all package pins at V_{CC} or GND, and the LCA configured with a MakeBits tie option.

CLB Switching Characteristic Guidelines



1105 26

Buffer (Internal) Switching Characteristic Guidelines

Description	Symbol	-70 Max	-100 Max	-125 Max	Units
Global and Alternate Clock Distribution*					
Either: **Normal** IOB input pad through clock buffer to any CLB or IOB clock input	T_{PID}	8.0	7.5	7.0	ns
Or: **Fast** (CMOS only) input pad through clock buffer to any CLB or IOB clock input	T_{PIDC}	6.5	6.0	5.7	ns
TBUF driving a Horizontal Longline (L.L.)*					
I to L.L. while T is Low (buffer active)	T_{IO}	5.0	4.7	4.5	ns
T↓ to L.L. active and valid with single pull-up resistor	T_{ON}	11.0	10.0	9.0	ns
T↓ to L.L. active and valid with pair of pull-up resistors	T_{ON}	12.0	11.0	10.0	ns
T↑ to L.L. High with single pull-up resistor	T_{PUS}	24.0	22.0	17.0	ns
T↑ to L.L. High with pair of pull-up resistors	T_{PUF}	17.0	15.0	12.0	ns
BIDI					
Bidirectional buffer delay	T_{BIDI}	2.0	1.8	1.7	ns

* Timing is based on the XC3042, for other devices see XACT timing calculator.

CLB Switching Characteristic Guidelines (continued)

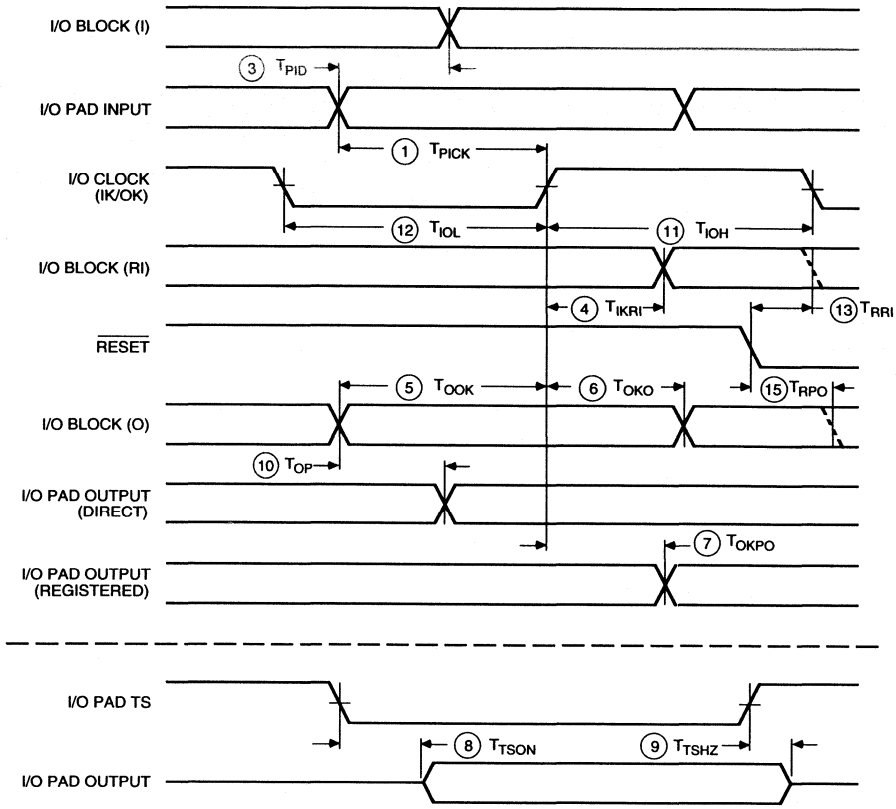
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

Speed Grade			-70		-100		-125		
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Combinatorial Delay									
Logic Variables A, B, C, D, E, to outputs X or Y	1	T_{ILO}		9.0		7.0		5.5	ns
Sequential delay									
Clock k to outputs X or Y	8	T_{CKO}		6.0		5.0		4.5	ns
Clock k to outputs X or Y when Q is returned through function generators F or G to drive X or Y		T_{QLO}		13.0		10.0		8.0	ns
Set-up time before clock K									
Logic Variables A, B, C, D, E	2	T_{ICK}	8.0		7.0		5.5		ns
Data In DI	4	T_{DICK}	5.0		4.0		3.0		ns
Enable Clock EC	6	T_{ECCK}	7.0		5.0		4.5		ns
Reset Direct inactive RD			1.0		1.0		1.0		ns
Hold Time after clock K									
Logic Variables A, B, C, D, E	3	T_{CKI}	0		0		0		ns
Data In DI	5	T_{CKDI}	4.0		2.0		1.5		ns
Enable Clock EC	7	T_{CKEC}	0		0		0		ns
Clock									
Clock High time	11	T_{CH}	5.0		4.0		3.0		ns
Clock Low time	12	T_{CL}	5.0		4.0		3.0		ns
Max flip-flop toggle rate		F_{CLK}	70		100		125		MHz
Reset Direct (RD)									
RD width	13	T_{RPW}	8.0		7.0		6.0		ns
delay from rd to outputs X or Y	9	T_{RIO}		8.0		7.0		6.0	ns
Global Reset ($\overline{\text{RESET}}$ Pad)*									
$\overline{\text{RESET}}$ width (Low)		T_{MRW}	25.0		21.0		20.0		ns
delay from $\overline{\text{RESET}}$ pad to outputs X or Y		T_{MRQ}		23.0		19.0		17.0	ns

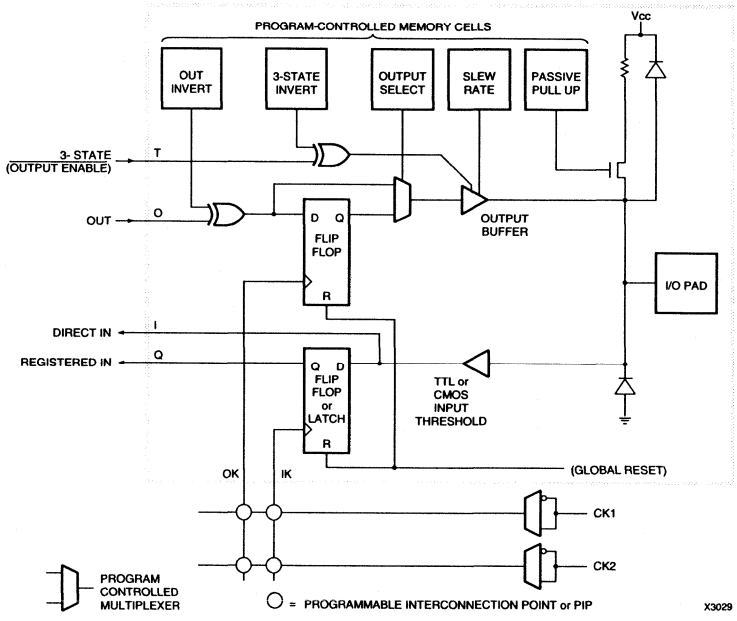
*Timing is based on the XC3042, for other devices see XACT timing calculator.

Note: The CLB K to Q output delay (T_{CKO}, #8) of any CLB, plus the shortest possible interconnect delay, is always longer than the Data In hold time requirement (T_{CKDI}, #5) of any CLB on the same die.

IOB Switching Characteristic Guidelines



1105 27C



IOB Switching Characteristic Guidelines (continued)

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-70		-100		-125		Units
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	
Propagation Delays (Input)									
Pad to Direct In (I)	3	T_{PID}		6		4		3	ns
Pad to Registered In (Q) with latch transparent		T_{PTG}		21		17		16	ns
Clock (IK) to Registered In (Q)	4	T_{IKRI}		5.5		4		3	ns
Set-up Time (Input)									
Pad to Clock (IK) set-up time	1	T_{PICK}	20		17		16		ns
Propagation Delays (Output)									
Clock (OK) to Pad (fast)	7	T_{OKPO}		13		10		9	ns
same (slew rate limited)	7	T_{OKPO}		33		27		24	ns
Output (O) to Pad (fast)	10	T_{OPF}		9		6		5	ns
same (slew-rate limited)	10	T_{OPS}		29		23		20	ns
3-state to Pad begin hi-Z (fast)	9	T_{TSHZ}		8		8		7	ns
same (slew-rate limited)	9	T_{TSHZ}		28		25		24	ns
3-state to Pad active and valid (fast)	8	T_{TSON}		14		12		11	ns
same (slew -rate limited)	8	T_{TSON}		34		29		27	ns
Set-up and Hold Times (Output)									
Output (O) to clock (OK) set-up time	5	T_{OOK}	10		9		8		ns
Output (O) to clock (OK) hold time	6	T_{OKO}	0		0		0		ns
Clock									
Clock High time	11	T_{IOH}	5		4		3		ns
Clock Low time	12	T_{IOL}	5		4		3		ns
Max. flip-flop toggle rate		F_{CLK}	70		100		125		MHz
Global Reset Delays (based on XC3042)									
$\overline{\text{RESET}}$ Pad to Registered In (Q)	13	T_{RRI}		25		24		23	ns
$\overline{\text{RESET}}$ Pad to output pad (fast)	15	T_{RPO}		35		33		29	ns
(slew-rate limited)	15	T_{RPO}		53		45		42	ns

Notes: 1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture). For larger capacitive loads, see XAPP 024. Typical slew rate limited output rise/fall times are approximately four times longer.

2. Voltage levels of unused (bonded and unbonded) pads must be valid logic levels. Each can be configured with the internal pull-up resistor or alternatively configured as a driven output or driven from an external source.

3. Input pad set-up time is specified with respect to the internal clock (IK). In order to calculate system set-up time, subtract clock delay (pad to IK) from the input pad set-up time value. Input pad holdtime with respect to the internal clock (IK) is negative. This means that pad level changes immediately before the internal clock edge (IK) will not be recognized.

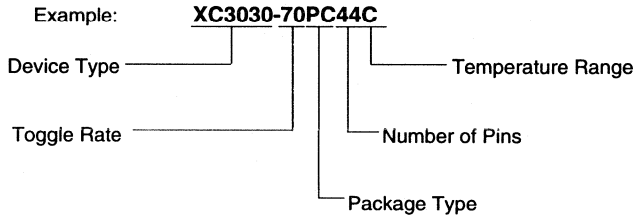
For a detailed description of the device architecture, see pages 2-105 through 2-123.

For a detailed description of the configuration modes and their timing, see pages 2-124 through 2-132.

For detailed lists of package pin-outs, see pages 2-140 through 2-150.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS		44	64	68	84		100				132		144	160	164	175		176	208	223
TYPE		PLAST. PLCC	PLAST. VQFP	PLAST. PLCC	PLAST. PLCC	CERAM. PGA	PLAST. PQFP	PLAST. TQFP	PLAST. VQFP	TOP-BRAZED CQFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	TOP-BRAZED CQFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	CERAM. PGA
CODE		PC44	VQ64	PC68	PC84	PG84	PQ100	TQ100	VQ100	CB100	PP132	PG132	TQ144	PQ160	CB164	PP175	PG175	TQ176	PQ208	PG223
XC3020	-50					MB				MB										
	-70			CI	CI	CIMB	CI			CMB										
	-100			CI	CI	CIMB	CI			CMB										
	-125			C	C	C	C													
XC3030	-50					M														
	-70	CI		CI	CI	CIM	CI	C												
	-100	CI		CI	CI	CIM	CI	C												
	-125	C		C	C	C	C	C												
XC3042	-50					MB				MB		MB								
	-70				CI	CIMB	CI	C		CMB	C	CIMB								
	-100				CI	CIMB	CI	C		CMB	C	CIMB								
	-125				C	C	C	C			C	C								
XC3064	-50											M								
	-70				CI						CI	CIM		CI						
	-100				CI						CI	CIM		CI						
	-125				C						C	C		C						
XC3090	-50														MB		MB			
	-70				CI									CI	CMB	CI	CIMB		CI	
	-100				CI									CI	CMB	CI	CIMB		CI	
	-125				C									C		C	C		C	

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C B = MIL-STD-883C Class B

Parentheses indicate future product plans

XC3000A
Logic Cell Array Family

Product Specifications

Features

- Enhanced, high performance FPGA family with five device types
  - Improved redesign of the basic XC3000 LCA Family
  - Logic densities from 1,000 to 6,000 gates
  - Up to 144 user-definable I/Os
- Superset of the industry-leading XC3000 family
  - Identical to the basic XC3000 in structure, pin out, design methodology, and software tools
  - 100% compatible with all XC3000, XC3000L, and XC3100 bitstreams
  - Improved routing and additional features
- Additional programmable interconnection points (PIPs)
  - Improved access to longlines and CLB clock enable inputs
  - Most efficient XC3000-class solution to bus-oriented designs
- Advanced 0.8 µ CMOS static memory technology
  - Low quiescent and active power consumption
- Performance specified by logic delays, faster than corresponding XC3000 versions
- XC3000A-specific features
  - 4 mA output sink and source current
  - Error checking of the configuration bitstream
  - Soft startup starts all outputs in slew-limited mode upon power-up
  - Easy migration to the XC3400 series of HardWire mask programmed devices for high-volume production.

Description

The XC3000A family offers the following enhancements over the popular XC3000 family:

The XC3000A family has additional interconnect resources to drive the I-inputs of TBUFs driving horizontal Longlines. The CLB Clock Enable input can be driven from a second vertical Longline. These two additions result in more efficient and faster designs when horizontal Longlines are used for data bussing.

During configuration, the XC3000A devices check the bitstream format for stop bits in the appropriate positions. Any error terminates the configuration and pulls INIT Low.

When the configuration process is finished and the device starts up in user mode, the first activation of the outputs is automatically slew-rate limited . This feature, called Soft Startup, avoids the potential ground bounce when all outputs are turned on simultaneously. After start-up, the slew rate of the individual outputs is, as in the XC3000 family, determined by the individual configuration option.

The XC3000A family is a superset of the XC3000 family. Any bitstream used to configure an XC3000 or XC3100 device configures an XC3000A device exactly the same way.

Device	CLBs	Array	User I/Os Max	Flip-Flops	Horizontal Longlines	Configurable Data Bits
XC3020A	64	8 x 8	64	256	16	14,779
XC3030A	100	10 x 10	80	360	20	22,176
XC3042A	144	12 x 12	96	480	24	30,784
XC3064A	224	16 x 14	120	688	32	46,064
XC3090A	320	16 x 20	144	928	40	64,160

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to Vcc +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to Vcc +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C
T_J	Junction temperature plastic	+125	°C
	Junction temperature ceramic	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

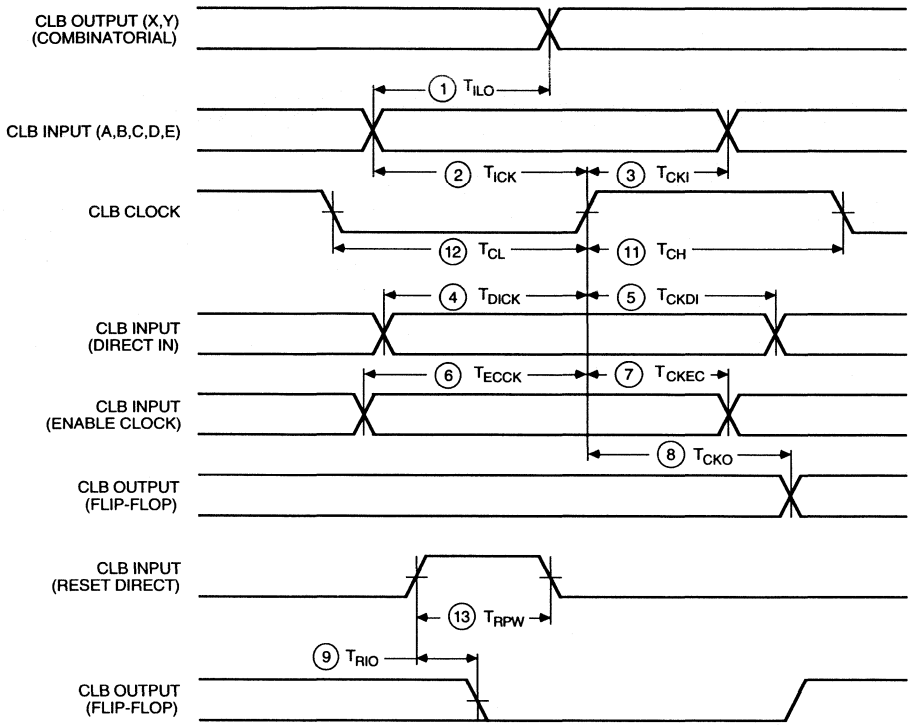
Symbol	Description	Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial 0°C to +70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial –40°C to +85°C	4.5	5.5	V
V_{IHT}	High-level input voltage — TTL configuration	2.0	Vcc	V
V_{ILT}	Low-level input voltage — TTL configuration	0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration	70%	100%	V_{CC}
V_{ILC}	Low-level input voltage — CMOS configuration	0	20%	V_{CC}
T_{IN}	Input signal transition time		250	ns

DC Characteristics Over Operating Conditions

Symbol	Description		Min	Max	Units
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 mA, V_{CC} min)	Commercial	3.86		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} max)			0.40	V
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 mA, V_{CC} min)	Industrial	3.76		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} max)			0.40	V
V_{CCPD}	Power-down supply voltage (PWRDWN must be Low)		2.30		V
I_{CCPD}	Power-down supply current ($V_{CC(MAX)}$ @ T_{MAX})	XC3020A		50	µA
		XC3030A		80	µA
		XC3042A		120	µA
		XC3064A		170	µA
		XC3090A		250	µA
I_{CCO}	Quiescent LCA supply current in addition to I_{CCPD}* Chip thresholds programmed as CMOS levels			500	µA
	Chip thresholds programmed as TTL levels			10	mA
I_{IL}	Input Leakage Current		–10	+10	µA
C_{IN}	Input capacitance, all packages except PGA175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			 10 15	 pF pF
	Input capacitance, PGA 175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			 15 20	 pF pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0 V (sample tested)		0.02	0.17	mA
I_{RLL}	Horizontal Longline pull-up (when selected) @ logic Low			3.4	mA

* With no output current loads, no active input or Longline pull-up resistors, all package pins at Vcc or GND, and the LCA device configured with a MakeBits tie option.

CLB Switching Characteristic Guidelines



1105 26

Buffer (Internal) Switching Characteristic Guidelines

Description	Symbol	-7 Max	-6 Max		Units
	Speed Grade	**-7**	**-6**		
Global and Alternate Clock Distribution*					
Either: **Normal** IOB input pad through clock buffer to any CLB or IOB clock input	T_{PID}	7.5	7.0		ns
Or: **Fast** (CMOS only) input pad through clock buffer to any CLB or IOB clock input	T_{PIDC}	6.0	5.7		ns
TBUF driving a Horizontal Longline (L.L.)*					
I to L.L. while T is Low (buffer active)	T_{IO}	4.5	4.0		ns
T↓ to L.L. active and valid with single pull-up resistor	T_{ON}	9.0	8.0		ns
T↓ to L.L. active and valid with pair of pull-up resistors	T_{ON}	11.0	10.0		ns
T↑ to L.L. High with single pull-up resistor	T_{PUS}	16.0	14.0		ns
T↑ to L.L. High with pair of pull-up resistors	T_{PUF}	10.0	8.0		ns
BIDI					
Bidirectional buffer delay	T_{BIDI}	1.7	1.5		ns

* Timing is based on the XC3042A, for other devices see XACT timing calculator.

CLB Switching Characteristic Guidelines (continued)

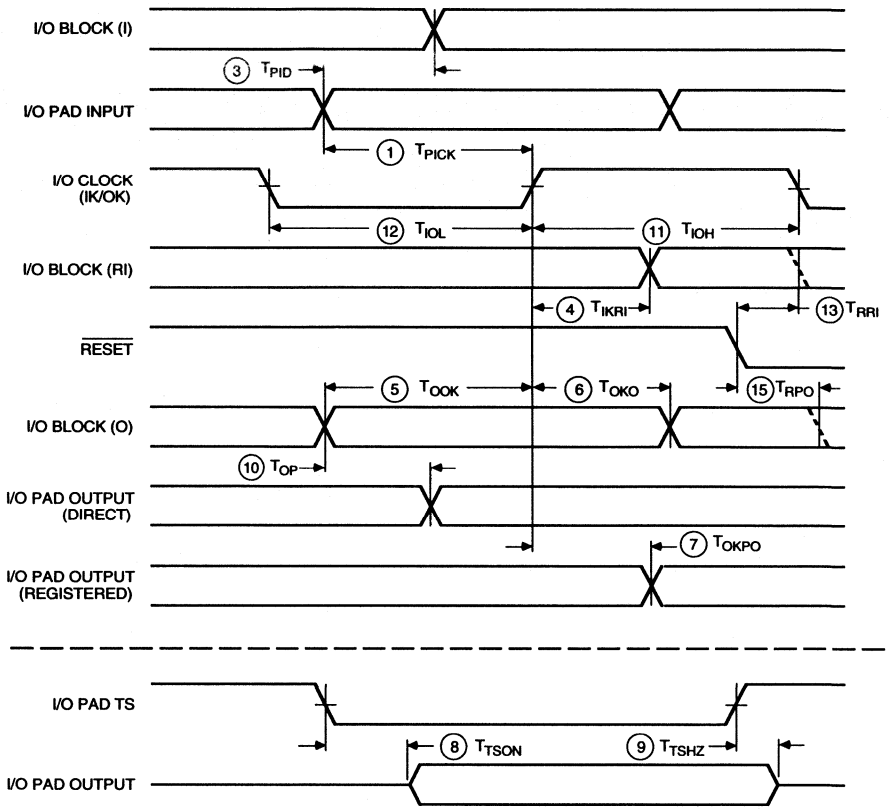
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-7		-6				
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**			**Units**
Combinatorial Delay									
Logic Variables A, B, C, D, E, to outputs X or Y									
FG Mode	1	T_{ILO}		5.1		4.1			ns
F and FGM Mode				5.6		4.6			ns
Sequential delay									
Clock k to outputs X or Y	8	T_{CKO}		4.5		3.5			ns
Clock k to outputs X or Y when Q is returned through function generators F or G to drive X or Y									
FG Mode		T_{QLO}		9.5		8.0			ns
F and FGM Mode				10.0		8.5			ns
Set-up time before clock K									
Logic Variables A, B, C, D, E									
FG Mode	2	T_{ICK}	4.5		3.5				ns
F and FGM Mode			5.0		4.0				ns
Data In DI	4	T_{DICK}	4.0		3.0				ns
Enable Clock EC	6	T_{ECCK}	4.5		4.0				ns
Hold Time after clock K									
Logic Variables A, B, C, D, E	3	T_{CKI}	0		0				ns
Data In DI	5	T_{CKDI}	1.0		1.0				ns
Enable Clock EC	7	T_{CKEC}	2.0		2.0				ns
Clock									
Clock High time	11	T_{CH}	4.0		3.5				ns
Clock Low time	12	T_{CL}	4.0		3.5				ns
Max. flip-flop toggle rate		F_{CLK}	113.0		135.0				MHz
Reset Direct (RD)									
RD width	13	T_{RPW}	6.0		5.0				ns
delay from RD to outputs X or Y	9	T_{RIO}		6.0		5.0			ns
Global Reset ($\overline{\text{RESET}}$ Pad)*									
$\overline{\text{RESET}}$ width (Low)		T_{MRW}	16.0		14.0				ns
delay from $\overline{\text{RESET}}$ pad to outputs X or Y		T_{MRQ}		19.0		17.0			ns

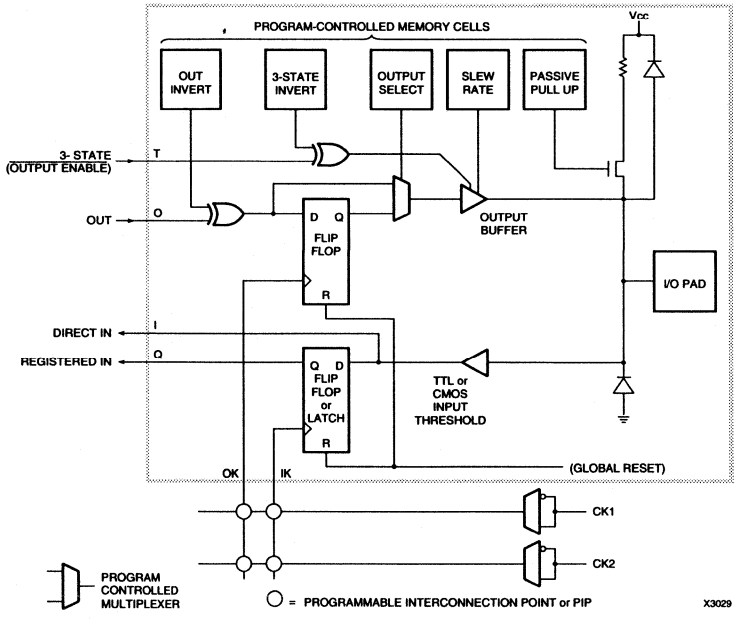
*Timing is based on the XC3042A, for other devices see XACT timing calculator.

Notes: The CLB K to Q output delay (T_{CKO}, #8) of any CLB, plus the shortest possible interconnect delay, is always longer than the Data In hold time requirement (T_{CKDI}, #5) of any CLB on the same die.

IOB Switching Characteristic Guidelines



1105 27C



IOB Switching Characteristic Guidelines (continued)

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-7		-6				
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**			**Units**
Propagation Delays (Input)									
Pad to Direct In (I)	3	T_{PID}		4.0		3.0			ns
Pad to Registered In (Q) with latch transparent		T_{PTG}		15.0		14.0			ns
Clock (IK) to Registered In (Q)	4	T_{IKRI}		3.0		2.5			ns
Set-up Time (Input)									
Pad to Clock (IK) set-up time	1	T_{PICK}	14.0		12.0				ns
Propagation Delays (Output)									
Clock (OK) to Pad (fast)	7	T_{OKPO}		8.0		7.0			ns
same (slew rate limited)	7	T_{OKPO}		18.0		15.0			ns
Output (O) to Pad (fast)	10	T_{OPF}		6.0		5.0			ns
same (slew-rate limited)	10	T_{OPS}		16.0		13.0			ns
3-state to Pad begin hi-Z (fast)	9	T_{TSHZ}		10.0		9.0			ns
same (slew-rate limited)	9	T_{TSHZ}		20.0		12.0			ns
3-state to Pad active and valid (fast)	8	T_{TSON}		11.0		10.0			ns
same (slew -rate limited)	8	T_{TSON}		21.0		18.0			ns
Set-up and Hold Times (Output)									
Output (O) to clock (OK) set-up time	5	T_{OOK}	8.0		7.0				ns
Output (O) to clock (OK) hold time	6	T_{OKO}	0		0				ns
Clock									
Clock High time	11	T_{IOH}	4.0		3.5				ns
Clock Low time	12	T_{IOL}	4.0		3.5				ns
Max. flip-flop toggle rate		F_{CLK}	113.0		135.0				MHz
Global Reset Delays (based on XC3042A)									
$\overline{\text{RESET}}$ Pad to Registered In (Q)	13	T_{RRI}		24.0		23.0			ns
$\overline{\text{RESET}}$ Pad to output pad (fast)	15	T_{RPO}		33.0		29.0			ns
(slew-rate limited)	15	T_{RPO}		43.0		37.0			ns

Notes:
1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture). For larger capacitive loads, see page XAPP024. Typical slew rate limited output rise/fall times are approximately four times longer.
2. Voltage levels of unused (bonded and unbonded) pads must be valid logic levels. Each can be configured with the internal pull-up resistor or alternatively configured as a driven output or driven from an external source.
3. Input pad set-up time is specified with respect to the internal clock (IK). In order to calculate system set-up time, subtract clock delay (pad to IK) from the input pad set-up time value. Input pad holdtime with respect to the internal clock (IK) is negative. This means that pad level changes immediately before the internal clock edge (IK) will not be recognized.
4. T_{PID}, T_{PTG}, and T_{PICK} are 3 ns higher for XTL2 when the pin is configured as a user input.

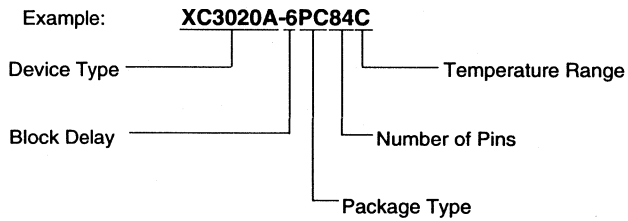
For a detailed description of the device architecture, see pages 2-105 through 2-123.

For a detailed description of the configuration modes and their timing, see pages 2-124 through 2-132.

For detailed lists of package pin-outs, see pages 2-140 through 2-150.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS		44	64	68	84		100				132		144	160	164	175		176	208	223
TYPE		PLAST. PLCC	PLAST. VQFP	PLAST. PLCC	PLAST. PLCC	CERAM PGA	PLAST. PQFP	PLAST. TQFP	PLAST. VQFP	TOP-BRAZED COFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	TOP-BRAZED COFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	CERAM. PGA
CODE		PC44	VQ64	PC68	PC84	PG84	PQ100	TQ100	VQ100	CB100	PP132	PG132	TQ144	PQ160	CB164	PP175	PG175	TQ176	PQ208	PG223
XC3020A	-7			C I	C I	C I	C I													
	-6			C	C	C	C													
XC3030A	-7	C I	C I	C I	C I	C I	C I		C I											
	-6	C	C	C	C	C	C		C											
XC3042A	-7				C I	C I	C I		C I		C I	C I	C I							
	-6				C	C	C		C		C	C	C							
XC3064A	-7				C I						C I	C I	C I	C I						
	-6				C						C	C	C	C						
XC3090A	-7				C I									C I		C I	C I	C I	C I	
	-6				C									C		C	C	C	C	

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C B = MIL-STD-883C Class B

XC3000L Low Voltage Logic Cell Array Family

Product Specifications

Features

- Part of the ZERO+ family of 3.3 V FPGAs
- Low supply voltage FPGA family with five device types
  - JEDEC-compliant 3.3 V version of theXC3000A LCA Family
  - Logic densities from 1,000 to 6,000 gates
  - Up to 144 user-definable I/Os
- Advanced, low power 0.8 µ CMOS static memory technology
  - Very low quiescent current consumption, ≤ 20µA
  - Operating power consumption 56% less than XC3000A, 66% less than previous generation 5 V FPGAs
- Superset of the industry-leading XC3000 family
  - Identical to the basic XC3000 in structure, pinout, design methodology, and software tools
  - 100% compatible with all XC3000, XC3000A, XC3100 and XC3100A bitstreams
  - Improved routing and additional features
- Additional programmable interconnection points (PIPs)
  - Improved access to Longlines and CLB clock enable inputs
  - Most efficient XC3000-class solution to bus-oriented designs
- XC3000L-specific features
  - Guaranteed over the 3.0 to 3.6 V Vcc range
  - 4 mA output sink and source current
  - Error checking of the configuration bitstream
  - Soft startup starts all outputs in slew-limited mode upon power-up
  - Easy migration to the XC3400 series of HardWire mask programmed devices for high-volume production

Description

The XC3000L family of FPGAs is optimized for operation from a nominally 3.3 V supply. Aside from the electrical and timing parameters listed in this data sheet, the XC3000L family is in all respects identical with the XC3000A family, and is a superset of the XC3000 family.

The operating power consumption of Xilinx FPGAs is almost exclusively dynamic, and it changes with the square of the supply voltage. For a given complexity and clock speed, the XC3000L consumes, therefore, only 44% of the power used by the equivalent XC3000A device. In accordance with its use in battery-powered equipment, the XC3000L family was designed for the lowest possible power-down and quiescent current consumption.

In mixed supply-voltage systems, the XC3000L, fed by a 3.3 V (nominal) supply, can directly drive any device with TTL-like input thresholds. When a 5 V device drives the XC3000L, a current-limiting resistor (1 kΩ) or a voltage divider is required to prevent excessive input current.

Like the XC3000A family, XC3000L offers the following functional improvements over the popular XC3000 family:

The XC3000L family has additional interconnect resources to drive the I-inputs of TBUFs driving horizontal Longlines. The CLB Clock Enable input can be driven from a second vertical Longline. These two additions result in more efficient and faster designs when horizontal Longlines are used for data bussing.

During configuration, the XC3000L devices check the bitstream format for stop bits in the appropriate positions. Any error terminates the configuration and pulls INIT Low.

When the configuration process is finished and the device starts up in user mode, the first activation of the outputs is automatically slew-rate limited. This feature, called Soft Startup, avoids the potential ground bounce when all outputs are turned on simultaneously. After start-up, the slew rate of the individual outputs is, as in the XC3000 family, determined by the individual configuration option.

The XC3000L family is a superset of the XC3000 family. Any bitstream used to configure an XC3000 device configures an XC3000L device the same way.

Device	CLBs	Array	User I/Os Max	Flip-Flops	Horizontal Longlines	Configurable Data Bits
XC3020L	64	8 x 8	64	256	16	14,779
XC3030L	100	10 x 10	80	360	20	22,176
XC3042L	144	12 x 12	96	480	24	30,784
XC3064L	224	16 x 14	120	688	32	46,064
XC3090L	320	16 x 20	144	928	40	64,160

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C
T_J	Junction temperature plastic	+125	°C
	Junction temperature ceramic	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description	Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial 0°C to +70°C	3.0	3.6	V
V_{IH}	High-level input voltage	2.0	V_{CC} +0.3	V
V_{IL}	Low-level input voltage	-0.3	0.8	V
T_{IN}	Input signal transition time		250	ns

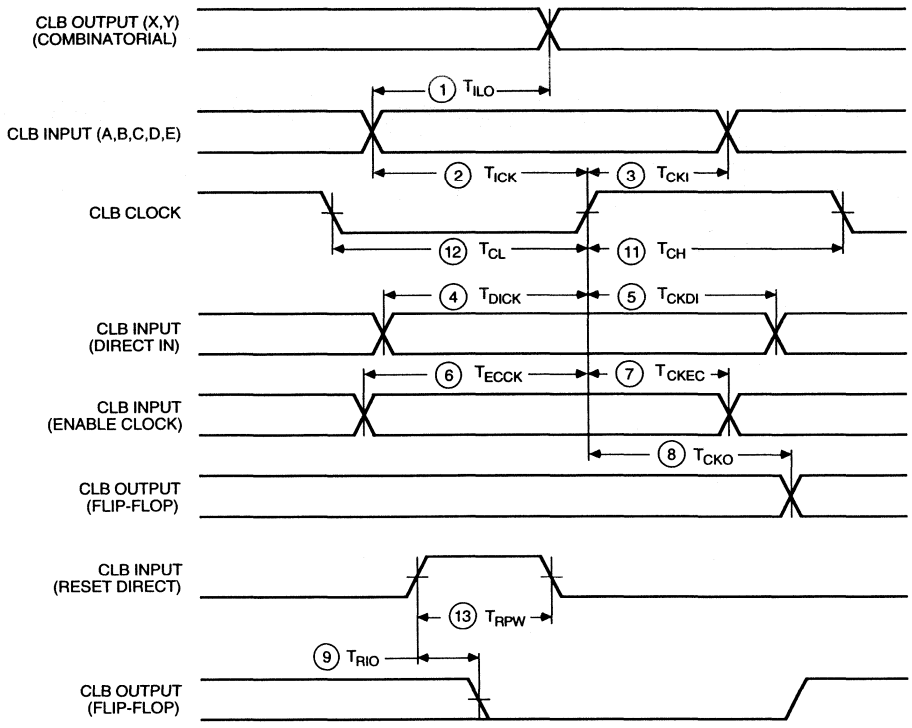
Although the present (1994) devices operate over the full supply voltage range from 3.0 to 5.25 V, Xilinx reserves the right to restrict operation to the 3.0 to 3.6 V range later, when smaller device geometries might preclude operation at 5 V.

DC Characteristics Over Operating Conditions

Symbol	Description	Min	Max	Units
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 mA, V_{CC} min)	2.40		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA, V_{CC} max)		0.40	V
V_{OH}	High-level output voltage (@ –100 µA, V_{CC} min)	V_{CC} -0.2		V
V_{OL}	Low-level output voltage (@ 100 µA, V_{CC} max)		0.2	V
V_{CCPD}	Power-down supply voltage ($\overline{PWRDWN}$ must be Low)	2.30		V
I_{CCPD}	Power-down supply current ($V_{CC(MAX)}$ @ T_{MAX})		10	µA
I_{CCO}	Quiescent LCA supply current* Chip thresholds programmed as CMOS levels		20	µA
I_{IL}	Input Leakage Current, all I/O pins in parallel	–10	+10	µA
C_{IN}	Input capacitance, all packages except PGA175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2		 10 15	 pF pF
	Input capacitance, PGA 175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2		 15 20	 pF pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0 V (sample tested)	0.02	0.17	mA
I_{RLL}	Horizontal Longline pull-up (when selected) @ logic Low		2.50	mA

* With no output current loads, no active input or Longline pull-up resistors, all package pins at V_{CC} or GND, and the LCA device configured with a MakeBits tie option. I_{CCO} is in addition to I_{CCPD}.

CLB Switching Characteristic Guidelines



1105 26

Buffer (Internal) Switching Characteristic Guidelines

Description	Symbol	-8 Max			Units
	Speed Grade	**-8**			
Global and Alternate Clock Distribution*					
Either: Normal IOB input pad through clock buffer to any CLB or IOB clock input	T_{PGC}	9.0			ns
Or: Fast (CMOS only) input pad through clock buffer to any CLB or IOB clock input	T_{PGCC}	7 .0			ns
TBUF driving a Horizontal Longline (L.L.)*					
I to L.L. while T is Low (buffer active)	T_{IO}	5 .0			ns
T↓ to L.L. active and valid with single pull-up resistor	T_{ON}	12 .0			ns
T↑ to L.L. High with single pull-up resistor	T_{PUS}	24 .0			ns
BIDI					
Bidirectional buffer delay	T_{BIDI}	2 .0			ns

* Timing is based on the XC3042L, for other devices see XACT timing calculator.
Note: The use of two pull-up resistors per Longline, available on other XC3000 devices, is not a valid design option for XC3000L devices

CLB Switching Characteristic Guidelines (continued)

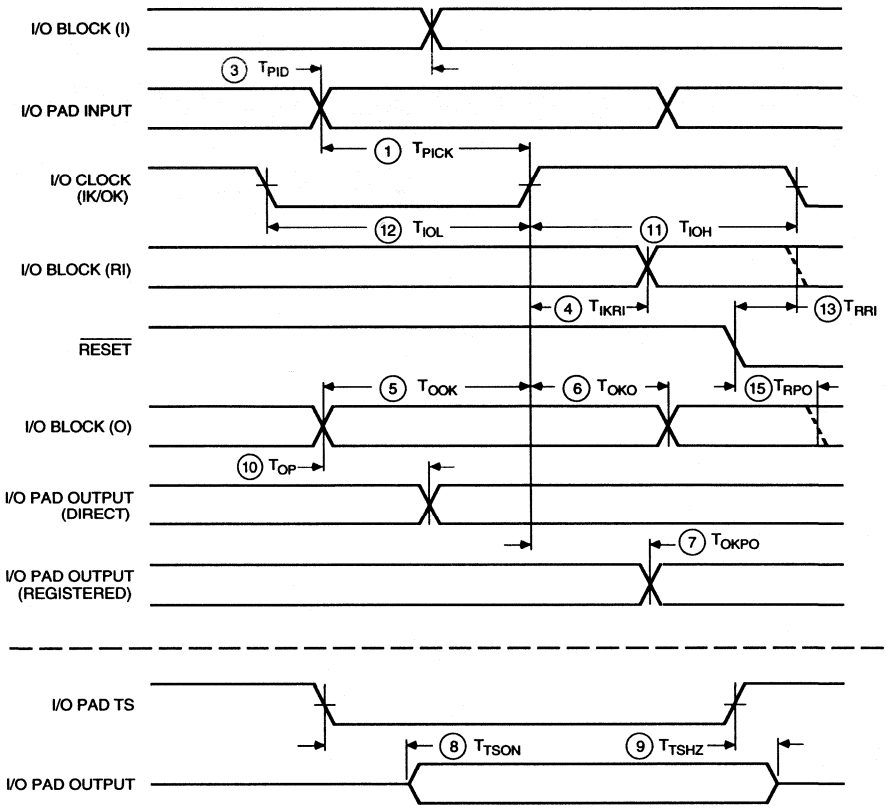
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

Speed Grade			-8						
Description	**Symbol**		**Min**	**Max**					**Units**
Combinatorial Delay									
Logic Variables A, B, C, D, E, to outputs X or Y									
FG Mode	1	T_{ILO}		6.7					ns
F and FGM Mode				7.5					ns
Sequential delay									
Clock k to outputs X or Y	8	T_{CKO}		7.5					ns
Clock k to outputs X or Y when Q is returned through function generators F or G to drive X or Y									
FG Mode		T_{QLO}		14.0					ns
F and FGM Mode				14.8					ns
Set-up time before clock K									
Logic Variables A, B, C, D, E									
FG MODE	2	T_{ICK}	5.0						ns
F and FGM Mode			5.8						ns
Data In DI	4	T_{DICK}	5.0						ns
Enable Clock EC	6	T_{ECCK}	5.0						ns
Hold Time after clock K									
Logic Variables A, B, C, D, E	3	T_{CKI}	0						ns
Data In DI	5	T_{CKDI}	2.0						ns
Enable Clock EC	7	T_{CKEC}	2.0						ns
Clock									
Clock High time	11	T_{CH}	5.0						ns
Clock Low time	12	T_{CL}	5.0						ns
Max. flip-flop toggle rate		F_{CLK}	80.0						MHz
Reset Direct (RD)									
RD width	13	T_{RPW}	7.0						ns
delay from RD to outputs X or Y	9	T_{RIO}	7.0						ns
Global Reset ($\overline{\text{RESET}}$ Pad)*									
$\overline{\text{RESET}}$ width (Low)		T_{MRW}	16.0						ns
delay from $\overline{\text{RESET}}$ pad to outputs X or Y		T_{MRQ}		23.0					ns

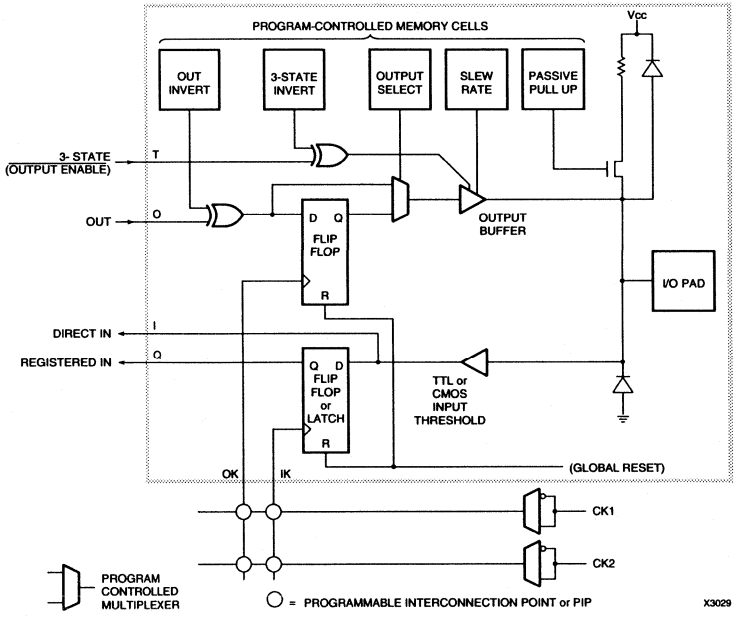
*Timing is based on the XC3042A, for other devices see XACT timing calculator.

Notes: The CLB K to Q output delay (T_{CKO}, #8) of any CLB, plus the shortest possible interconnect delay, is always longer than the Data In hold time requirement (T_{CKDI}, #5) of any CLB on the same die.

IOB Switching Characteristic Guidelines



1105 27C



IOB Switching Characteristic Guidelines (continued)

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-8						
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Propagation Delays (Input)									
Pad to Direct In (I)	3	T_{PID}		5.0					ns
Pad to Registered In (Q) with latch transparent		T_{PTG}		24.0					ns
Clock (IK) to Registered In (Q)	4	T_{IKRI}		6.0					ns
Set-up Time (Input)									
Pad to Clock (IK) set-up time	1	T_{PICK}	22.0						ns
Propagation Delays (Output)									
Clock (OK) to Pad (fast)	7	T_{OKPO}		12.0					ns
same (slew rate limited)	7	T_{OKPO}		28.0					ns
Output (O) to Pad (fast)	10	T_{OPF}		9.0					ns
same (slew-rate limited)	10	T_{OPS}		25.0					ns
3-state to Pad begin hi-Z (fast)	9	T_{TSHZ}		12.0					ns
same (slew-rate limited)	9	T_{TSHZ}		28.0					ns
3-state to Pad active and valid (fast)	8	T_{TSON}		16.0					ns
same (slew -rate limited)	8	T_{TSON}		32.0					ns
Set-up and Hold Times (Output)									
Output (O) to clock (OK) set-up time	5	T_{OOK}	12.0						ns
Output (O) to clock (OK) hold time	6	T_{OKO}	0						ns
Clock									
Clock High time	11	T_{IOH}	5.0						ns
Clock Low time	12	T_{IOL}	5.0						ns
Max. flip-flop toggle rate		F_{CLK}	80.0						MHz
Global Reset Delays (based on XC3042L)									
$\overline{\text{RESET}}$ Pad to Registered In (Q)	13	T_{RRI}	25.0						ns
$\overline{\text{RESET}}$ Pad to output pad (fast)	15	T_{RPO}	35.0						ns
(slew-rate limited)	15	T_{RPO}	51.0						ns

Notes:
1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture). For larger capacitive loads, see page XAPP024. Typical slew rate limited output rise/fall times are approximately four times longer.
2. Voltage levels of unused (bonded and unbonded) pads must be valid logic levels. Each can be configured with the internal pull-up resistor or alternatively configured as a driven output or driven from an external source.
3. Input pad set-up time is specified with respect to the internal clock (IK). In order to calculate system set-up time, subtract clock delay (pad to IK) from the input pad set-up time value. Input pad holdtime with respect to the internal clock (IK) is negative. This means that pad level changes immediately before the internal clock edge (IK) will not be recognized.
4. The slew-limited delays for T_{OKPO}, T_{SHZ}, T_{TSON}, and T_{RPO} are guaranteed by design and not tested.

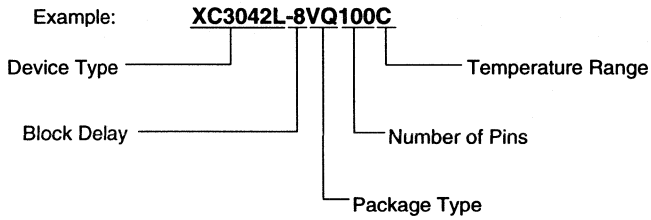
For a detailed description of the device architecture, see pages 2-105 through 2-123.

For a detailed description of the configuration modes and their timing, see pages 2-124 through 2-132.

For detailed lists of package pin-outs, see pages 2-140 through 2-150.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS	44	64	68	84		100				132		144	160	164	175		176	208	223
TYPE	PLAST. PLCC	PLAST. VQFP	PLAST. PLCC	PLAST. PLCC	CERAM PGA	PLAST. PQFP	PLAST. TQFP	PLAST. VQFP	TOP-BRAZED CQFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	TOP-BRAZED CQFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	CERAM. PGA
CODE	PC44	VQ64	PC68	PC84	PG84	PQ100	TQ100	VQ100	CB100	PP132	PG132	TQ144	PQ160	CB164	PP175	PG175	TQ176	PQ208	PG223
XC3020L				C															
XC3030L		C		C				C											
XC3042L				C				C				C							
XC3064L				C								C							
XC3090L				C													C		

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C B = MIL-STD-883C Class B

XC3100, XC3100A Logic Cell Array Families

Product Specifications

Features

- Two ultra-high-speed FPGA families with six members each
  - 50-80 MHz system clock rates
  - 190 to 270 MHz guaranteed flip-flop toggle rates
  - 2.7 to 4.1 ns logic delays
- High-end additional family members in the 22 X 22 CLB array-size XC3195 and XC3195A devices
- 8 mA output sink current and 8 mA source current
- Maximum power-down and quiescent current is 5 mA
- Both families are 100% architecture and pin-out compatible with other XC3000 families
- Beyond this, XC3100 is also software and bitstream compatible with the XC3000 family, while XC3100A is software and bitstream compatible with the XC3000A and XC3000L families

The XC3100A family is recommended for all new designs, since it offers improved functionality and enhanced development system support

XC3100A combines the features of the XC3000A and XC3100 families.

- Additional interconnect resources for TBUFs and CE inputs
- Error checking of the configuration bitstream
- Soft startup holds all outputs slew-rate limited during initial power-up
- More advanced CMOS process

Description

XC3100 and XC3100A are performance-optimized relatives of the XC3000 and XC3000A families. While all families are footprint compatible, the XC3100 and XC3100A families extend the system performance beyond 80 MHz.

The XC3100 and XC3100A families follow the XC4000 speed-grade nomenclature, indicating device performance with a number that is based on the internal logic-block delay, in ns.

The XC3100A family offers the following enhancements over the popular XC3100 family.

The XC3100A family has additional interconnect resources to drive the I-inputs of TBUFs driving horizontal Longlines. The CLB Clock Enable input can be driven from a second vertical Longline. These two additions result in more efficient and faster designs when horizontal Longlines are used for data bussing.

During configuration, the XC3100A devices check the bitstream format for stop bits in the appropriate positions. Any error terminates the configuration and pulls INIT Low.

When the configuration process is finished and the device starts up in user mode, the first activation of the outputs is automatically slew-rate limited. This feature, called Soft Startup, avoids the potential ground bounce when all outputs are turned on simultaneously. After start-up, the slew rate of the individual outputs is, as in all XC3000 families, determined by the individual configuration option.

The XC3100A family is a superset of the XC3000 families. Any bitstream used to configure an XC3000, XC3000A, XC3000L or XC3100 device, will configure the same-size XC3100A device exactly the same way.

Device	CLBs	Array	User I/O Max	Flip-Flops	Horizontal Longlines	Configuration Data Bits
XC3120 / XC3120A	64	8 x 8	64	256	16	14,779
XC3130 / XC3130A	100	10 x 10	80	360	20	22,176
XC3142 / XC3142A	144	12 x 12	96	480	24	30,784
XC3164 / XC3164A	224	16 x 14	120	688	28	46,064
XC3190 / XC3190A	320	16 x 20	144	928	40	64,160
XC3195 / XC3195A	484	22 x 22	176	1,320	44	94,944

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C
T_J	Junction temperature plastic	+125	°C
	Junction temperature ceramic	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description	Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial 0°C to +70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial –40°C to +85°C	4.5	5.5	V
V_{IHT}	High-level input voltage — TTL configuration	2.0	V_{CC}	V
V_{ILT}	Low-level input voltage — TTL configuration	0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration	70%	100%	V_{CC}
V_{ILC}	Low-level input voltage — CMOS configuration	0	20%	V_{CC}
T_{IN}	Input signal transition time		250	ns

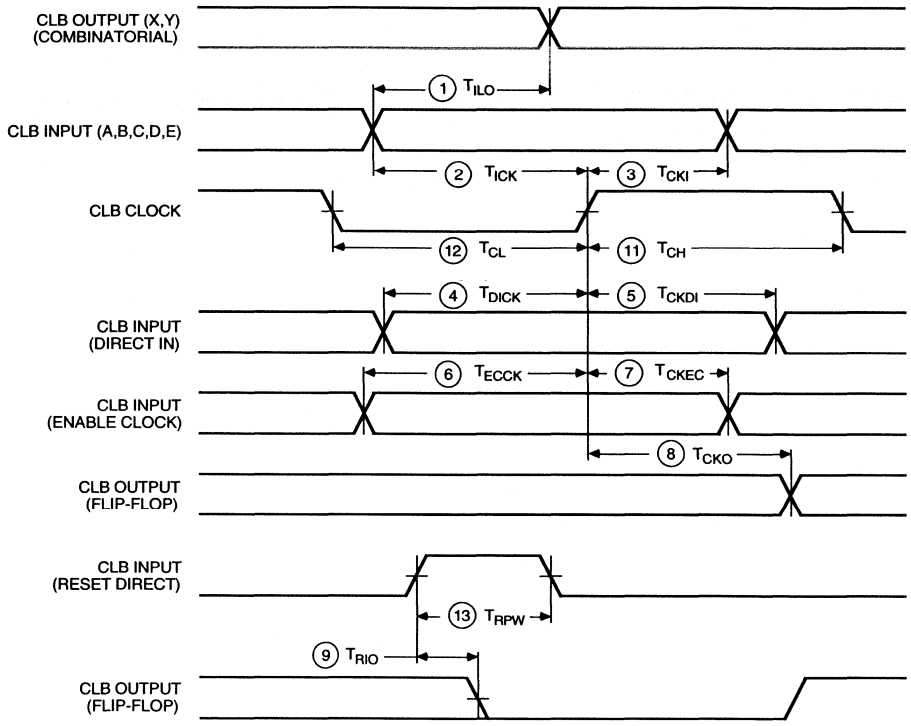
DC Characteristics Over Operating Conditions

Symbol	Description		Min	Max	Units
V_{OH}	High-level output voltage (@ I_{OH} = –8.0 mA, V_{CC} min)	Commercial	3.86		V
V_{OL}	Low-level output voltage (@ I_{OL} = 8.0 mA, V_{CC} max)			0.40	V
V_{OH}	High-level output voltage (@ I_{OH} = –8.0 mA, V_{CC} min)	Industrial	3.76		V
V_{OL}	Low-level output voltage (@ I_{OL} = 8.0 mA, V_{CC} max)			0.40	V
V_{CCPD}	Power-down supply voltage (PWRDWN must be Low)		2.30		V
I_{CCO}	Quiescent LCA supply current Chip thresholds programmed as CMOS levels[1]			5	mA
	Chip thresholds programmed as TTL levels			14	mA
I_{IL}	Input Leakage Current		–10	+10	µA
C_{IN}	Input capacitance, all packages except PGA175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			 10 15	 pF pF
	Input capacitance, PGA 175 (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			 15 20	 pF pF
I_{RIN}	Pad pull-up (when selected) @ V_{IN} = 0V (sample tested)		0.02	0.17	mA
I_{RLL}	Horizontal long line pull-up (when selected) @ logic Low		0.20	2.80	mA

Note: 1. With no output current loads, no active input or long line pull-up resistors, all package pins at V_{CC} or GND, and the LCA configured with a MakeBits tie option.

2. Total continuous output sink current may not exceed 100 mA per ground pin. The number of ground pins varies from two for the XC3120 in the PC84 package, to eight for the XC3195 in the PQ208 or PG223 package.

CLB Switching Characteristic Guidelines



1105 26

Buffer (Internal) Switching Characteristic Guidelines

	Speed Grade	-5	-4	-3	
Description	**Symbol**	**Max**	**Max**	**Max**	**Units**
Global and Alternate Clock Distribution*					
Either: **Normal** IOB input pad through clock buffer to any CLB or IOB clock input	T_{PID}	6.8	6.5	5.6	ns
Or: **Fast** (CMOS only) input pad through clock buffer to any CLB or IOB clock input	T_{PIDC}	5.4	5.1	4.3	ns
TBUF driving a Horizontal Long line (L.L.)*					
I to L.L. while T is Low (buffer active)	T_{IO}	4.1	3.7	3.1	ns
T↓ to L.L. active and valid with single pull-up resistor	T_{ON}	5.6	5.0	4.2	ns
T↓ to L.L. active and valid with pair of pull-up resistors	T_{ON}	7.1	6.5	5.7	ns
T↑ to L.L. High with single pull-up resistor	T_{PUS}	15.6	13.5	11.4	ns
T↑ to L.L. High with pair of pull-up resistors	T_{PUF}	12.0	10.5	8.8	ns
BIDI					
Bidirectional buffer delay	T_{BIDI}	1.4	1.2	1.0	ns

* Timing is based on the XC3142, for other devices see XACT timing calculator.

CLB Switching Characteristic Guidelines (continued)

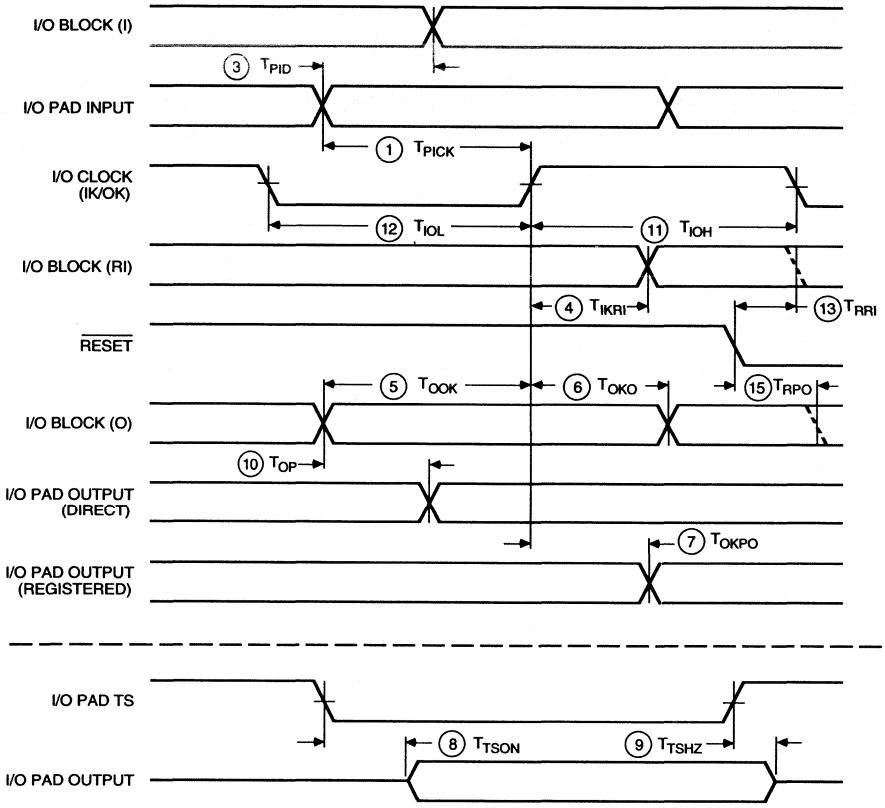
Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

	Speed Grade		-5		-4		-3		
Description	**Symbol**		**Min**	**Max**	**Min**	**Max**	**Min**	**Max**	**Units**
Combinatorial Delay Logic Variables A, B, C, D, E, to outputs X or Y	1	T_{ILO}		4.1		3.3		2.7	ns
Sequential delay Clock K to outputs X or Y Clock K to outputs X or Y when Q is returned through function generators F or G to drive X or Y	8	T_{CKO} T_{QLO}		3.1 6.3		2.5 5.2		2.1 4.3	ns ns
Set-up time before clock K Logic Variables A, B, C, D, E Data In DI Enable Clock EC Reset Direct inactive RD	2 4 6	T_{ICK} T_{DICK} T_{ECCK}	3.1 2.0 3.8 1.0		2.5 1.6 3.2 1.0		2.1 1.4 2.7 1.0		ns ns ns ns
Hold Time after clock k Logic Variables A, B, C, D, E Data In DI Enable Clock EC	3 5 7	T_{CKI} T_{CKDI} T_{CKEC}	0 1.2 1.0		0 1.0 0.8		0 0.9 0.7		ns ns ns
Clock Clock High time Clock Low time Max. flip-flop toggle rate	11 12	T_{CH} T_{CL} F_{CLK}	2.4 2.4 190		2.0 2.0 230		1.6 1.6 270		ns ns MHz
Reset Direct (RD) RD width delay from RD to outputs X or Y	13 9	T_{RPW} T_{RIO}	3.8	4.4	3.2	3.7	2.7	3.1	ns ns
Global Reset, from $\overline{\text{RESET}}$ Pad, based on XC3142 $\overline{\text{RESET}}$ width (Low) delay from $\overline{\text{RESET}}$ pad to outputs X or Y		T_{MRW} T_{MRQ}	18.0	17.0	15.0	14.0	13.0	12.0	ns ns

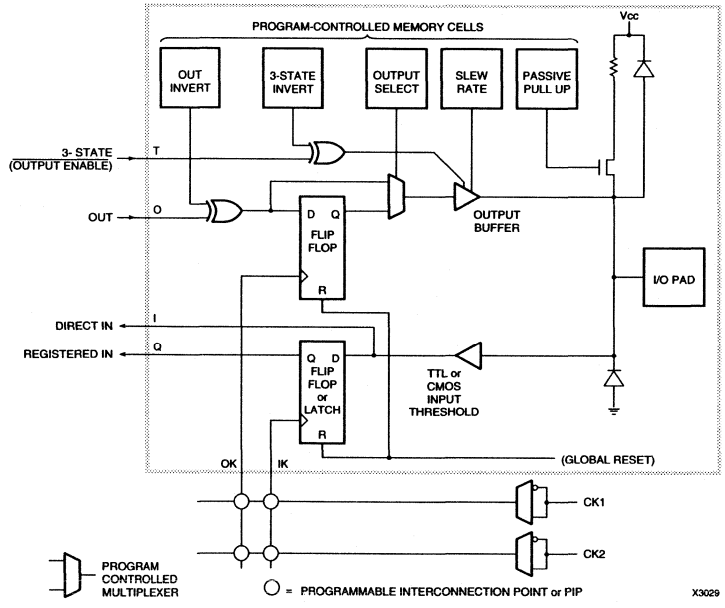
Notes: The CLB K to Q output delay (T_{CKO}, #8) of any CLB, plus the shortest possible interconnect delay, is always longer than the Data In hold time requirement (T_{CKDI}, #5) of any CLB on the same die.

T_{ILO}, T_{QLO} and T_{ICK} are specified for 4-input functions. For 5-input functions or base FGM functions, each of these specifications for the XC3100 family increases by 0.60 ns (-5), 0.6 ns (-4) and 0.5 ns (-3) and each of these specifications for the XC3100A family increses by 0.50 ns (-5), 0.42 ns (-4) and 0.35 ns (-3).

IOB Switching Characteristic Guidelines



1105 27C



IOB Switching Characteristic Guidelines (continued)

Testing of the switching parameters is modeled after testing methods specified by MIL-M-38510/605. All devices are 100% functionally tested. Since many internal timing parameters cannot be measured directly, they are derived from benchmark timing patterns. The following guidelines reflect worst-case values over the recommended operating conditions. For more detailed, more precise, and more up-to-date timing information, use the values provided by the XACT timing calculator and used in the simulator.

Description		Symbol		-5 Min	-5 Max	-4 Min	-4 Max	-3 Min	-3 Max	Units
Speed Grade										
Propagation Delays (Input)										
Pad to Direct In (I)		3	T_{PID}		2.8		2.5		2.2	ns
Pad to Registered In (q) with latch transparent			T_{PTG}		16.0		15.0		13.0	ns
Clock (IK) to Registered In (Q)		4	T_{IKRI}		2.8		2.5		2.2	ns
Set-up Time (Input)										
Pad to Clock (IK) set-up time	XC3100 Family	1	T_{PICK}	15.0		14.0		12.0		ns
	XC3120A,XC3130A			11.8		10.6		9.4		ns
	XC3142A			11.9		10.7		9.5		ns
	XC3164A			12.1		11.0		9.7		ns
	XC3190A			12.4		11.2		9.9		ns
	XC3195A			12.9		11.6		10.3		ns
Propagation Delays (Output)										
Clock (OK) to Pad	(fast)	7	T_{OKPO}		5.5		5.0		4.4	ns
same	(slew rate limited)	7	T_{OKPO}		14.0		12.0		10.0	ns
Output (O) to Pad	(fast)	10	T_{OPF}		4.1		3.7		3.3	ns
same	(slew-rate limited)	10	T_{OPS}		13.0		11.0		9.0	ns
3-state to Pad begin hi-Z	(fast)	9	T_{TSHZ}		6.9		6.2		5.5	ns
same	(slew-rate limited)	9	T_{TSHZ}		6.9		6.2		5.5	ns
3-state to Pad active and valid	(fast)	8	T_{TSON}		12.0		10.0		9.0	ns
same	(slew -rate limited)	8	T_{TSON}		20.0		17.0		15.0	ns
Set-up and Hold Times (Output)										
Output (O) to clock (OK) set-up time,	XC3100	5	T_{OOK}	6.2		5.6		5.0		ns
	XC3100A			5.0		4.5		4.0		ns
Output (O) to clock (OK) hold time		6	T_{OKO}	0		0		0		ns
Clock										
Clock High time		11	T_{IOH}	2.4		2.0		1.6		ns
Clock Low time		12	T_{IOL}	2.4		2.0		1.6		ns
Max. flip-flop toggle rate			F_{CLK}	190.0		230.0		270.0		MHz
Global Reset Delays (based on XC3142)										
$\overline{\text{RESET}}$ Pad to Registered In (Q),	XC3120/XC3120A	13	T_{RRI}		18.0		15.0		13.0	ns
	XC3195/XC3195A				29.5		25.0		21.0	ns
$\overline{\text{RESET}}$ Pad to output pad	(fast)	15	T_{RPO}		24.0		20.0		17.0	ns
	(slew-rate limited)	15	T_{RPO}		32.0		27.0		23.0	ns

Notes: 1. Timing is measured at pin threshold, with 50 pF external capacitive loads (incl. test fixture). For larger capacitive loads, see XAPP 024. Typical slew rate limited output rise/fall times are approximately four times longer.

2. Voltage levels of unused (bonded and unbonded) pads must be valid logic levels. Each can be configured with the internal pull-up resistor or alternatively configured as a driven output or driven from an external source.

3. Input pad set-up time is specified with respect to the internal clock (IK). In order to calculate system set-up time, subtract clock delay (pad to ik) from the input pad set-up time value. Input pad holdtime with respect to the internal clock (IK) is negative. This means that pad level changes immediately before the internal clock edge (IK) will not be recognized.

4. T_{PID}, T_{PTG}, and T_{PICK} are 3 ns higher for XTAL2 when the pin is configured as a user input.

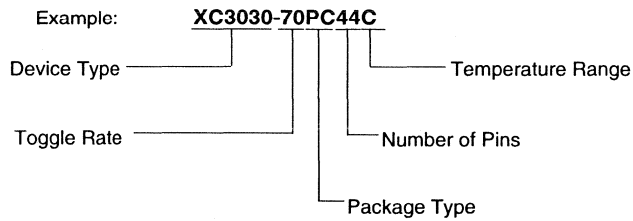
For a detailed description of the device architecture, see pages 2-105 through 2-123.

For a detailed description of the configuration modes and their timing, see pages 2-124 through 2-132.

For detailed lists of package pin-outs, see pages 2-140 through 2-150.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS		44	64	68	84		100				132		144	160	164	175		176	208	223
TYPE		PLAST. PLCC	PLAST. VQFP	PLAST. PLCC	PLAST. PLCC	CERAM PGA	PLAST. PQFP	PLAST. TQFP	PLAST. VQFP	TOP-BRAZED CQFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	TOP-BRAZED CQFP	PLAST. PGA	CERAM. PGA	PLAST. TQFP	PLAST. PQFP	CERAM. PGA
CODE		PC44	VQ64	PC68	PC84	PG84	PQ100	TQ100	VQ100	CB100	PP132	PG132	TQ144	PQ160	CB164	PP175	PG175	TQ176	PQ208	PG223
XC3120	-5			C I	C I	C I	C I (M B)			(M B)										
	-4			C I	C I	C I	C I													
	-3			C	C	C	C													
XC3130	-5	C I		C I	C I	C I	C I	C												
	-4	C I		C I	C I	C I	C I	C												
	-3	C		C	C	C	C	C												
XC3142	-5				C I	C I	C I (M B)	C		(M B)	C	C I (M B)	C I							
	-4				C I	C I	C I	C			C	C I	C I							
	-3				C	C	C	C			C	C	C							
XC3164	-5				C I						C I	C I		C I						
	-4				C I						C I	C I		C I						
	-3				C						C	C		C						
XC3190	-5				C I									C I	(M B)	C I	C I (M B)		C I	
	-4				C I									C I		C I	C I		C I	
	-3				C									C		C	C		C	
XC3195	-5				C I									C I		C I	C I (M B)		C I	C I (M B)
	-4				C I									C I		C I	C I		C I	C I
	-3				C									C		C	C		C	C
XC3120A	-5			C I	C I	C I	C I (M B)			(M B)										
	-4			C I	C I	C I	C I													
	-3			C	C	C	C													
XC3130A	-5	C I		C I	C I	C I	C I		C											
	-4	C I		C I	C I	C I	C I		C											
	-3	C		C	C	C	C		C											
XC3142A	-5				C I	C I	C I (M B)		C	(M B)	C	C I (M B)	C I							
	-4				C I	C I	C I		C		C	C I	C I							
	-3				C	C	C		C		C	C	C							
XC3164A	-5				C I						C I	C I		C I						
	-4				C I						C I	C I		C I						
	-3				C						C	C		C						
XC3190A	-5				C I									C I	(M B)	C I	C I (M B)		C I	
	-4				C I									C I		C I	C I		C I	
	-3				C									C		C	C		C	
XC3195A	-5				C I									C I		C I	C I (M B)		C I	C I (M B)
	-4				C I									C I		C I	C I		C I	C I
	-3				C									C		C	C		C	C

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C M = Mil Temp = -55° to +125° C B = MIL-STD-883C Class B

Parentheses indicate future product plans

XC2000 Logic Cell Array Families

Table of Contents

Overview

Introduced in 1985, the XC2000 family has seen continuously increasing sales for 8 years. In 1993, Xilinx introduced the ZERO+ Family of 3.3 V devices, intended for the fast growing market of battery-operated portable computers and instruments.

While the XC3000/XC3100 families offer more speed, a wider range of device capacities and more packaging options, and the XC4000 family offers more advanced systems features, the XC2064 and XC2018 are the world's lowest cost FPGAs, and they remain the most economical solution for all applications where the XC3020 or XC4002A features are not required.

XC2000
Logic Cell Array Families

Product Description

Features

- Fully Field-Programmable:
  - I/O functions
  - Digital logic functions
  - Interconnections
- General-purpose array architecture
- Complete user control of design cycle
- Compatible arrays with logic cell complexity equivalent from 600 to 1,500 gates
- Available in 5-V and 3.3-V versions
- 100% factory tested
- Selectable configuration modes
- Low-power, CMOS, static-memory technology
- Performance equivalent to TTL SSI/MSI
- TTL or CMOS input thresholds
- Complete development system support
  - XACT Design Editor
  - Schematic Entry
  - Macro Library
  - Timing Calculator
  - Logic and Timing Simulator
  - Auto Place / Route

Description

The Logic Cell Array (LCA) is a high density CMOS integrated circuit. Its user-programmable array architecture is made up of three types of configurable elements: Input/Output Blocks, logic blocks and Interconnect. The designer can define individual I/O blocks for interface to external circuitry, define logic blocks to implement logic functions and define interconnection networks to compose larger scale logic functions. The XACT Development System provides interactive graphic design capture and automatic routing. Both logic simulation and in-circuit emulation are available for design verification.

The Logic Cell Array is available in a variety of logic capacities, package styles, temperature ranges and speed grades.

Device	V_{CC}	Typ. Logic Capacity (gates)	CLBs	User I/O Max	Config. bits
XC2064	5.0 V	600 – 1,000	64	58	12,038
XC2064L	3.3 V	600 – 1,000	64	58	12,038
XC2018	5.0 V	1,000 - 1,500	100	74	17,878
XC2018L	3.3 V	1,000 - 1,500	100	74	17,878

The XC2000 family operates with a nominal 5.0 V supply. The XC2000L family operates with nominal 3.3 V supply.

The LCA logic functions and interconnections are determined by data stored in internal static-memory cells. On-chip logic provides for automatic loading of configuration data at power-up. The program data can reside in an EEPROM, EPROM or ROM on the circuit board or on a floppy disk or hard disk. The program can be loaded in a number of modes to accommodate various system requirements.

Architecture

The general structure of a Logic Cell Array is shown in Figure 1. The elements of the array include three categories of user programmable elements: I/O Blocks (IOBs), Configurable Logic Blocks (CLBs) and Programmable Interconnections. The I/OBs provide an interface between the logic array and the device package pins. The CLBs perform user-specified logic functions, and the interconnect resources are programmed to form networks that carry logic signals among the blocks.

LCA configuration is established through a distributed array of memory cells.The XACT development system generates the program used to configure the Logic Cell Array which includes logic to implement automatic configuration.

Configuration Memory

The configuration of the Logic Cell Array is established by programming memory cells which determine the logic functions and interconnections. The memory loading process is independent of the user logic functions.

The static memory cell used for the configuration memory in the Logic Cell Array has been designed specifically for high reliability and noise immunity. Based on this design, which has been patented, integrity of the LCA configuration memory is assured even under adverse conditions. Compared with other programming alternatives, static memory provides the best combination of high density, high performance, high reliability and comprehensive testability. As shown in Figure 2, the basic memory cell consists of two CMOS inverters plus a pass transistor used for writing data to the cell. The cell is only written during configuration and only read during readback. During normal operation the pass transistor is off and does not affect the stability of the cell. This is quite different from the normal operation of conventional memory devices, in which the cells are continuously read and rewritten.

The outputs Q and $\overline{Q}$ control pass-transistor gates directly. The absence of sense amplifiers and the output capacitive load provide additional stability to the cell. Due to the structure of the configuration memory cells, they are not affected by extreme power supply excursions or very high levels of alpha particle radiation. In reliability testing no soft errors have been observed, even in the presence of very high doses of alpha radiation.

Input/Output Block

Each user-configurable I/O block (IOB) provides an interface between the external package pin of the device and the internal logic. Each I/O block includes a programmable input path and a programmable output buffer. It also provides input clamping diodes to provide protection from electro-static damage, and circuits to protect the LCA from latch-up due to input currents. Figure 3 shows the general structure of the I/O block.

The input buffer portion of each I/O block provides threshold detection to translate external signals applied to the package pin to internal logic levels. The input buffer threshold of the I/O blocks can be programmed to be compatible with either TTL (1.4 V) or CMOS (2.2 V) levels. The buffered input signal drives both the data input of an

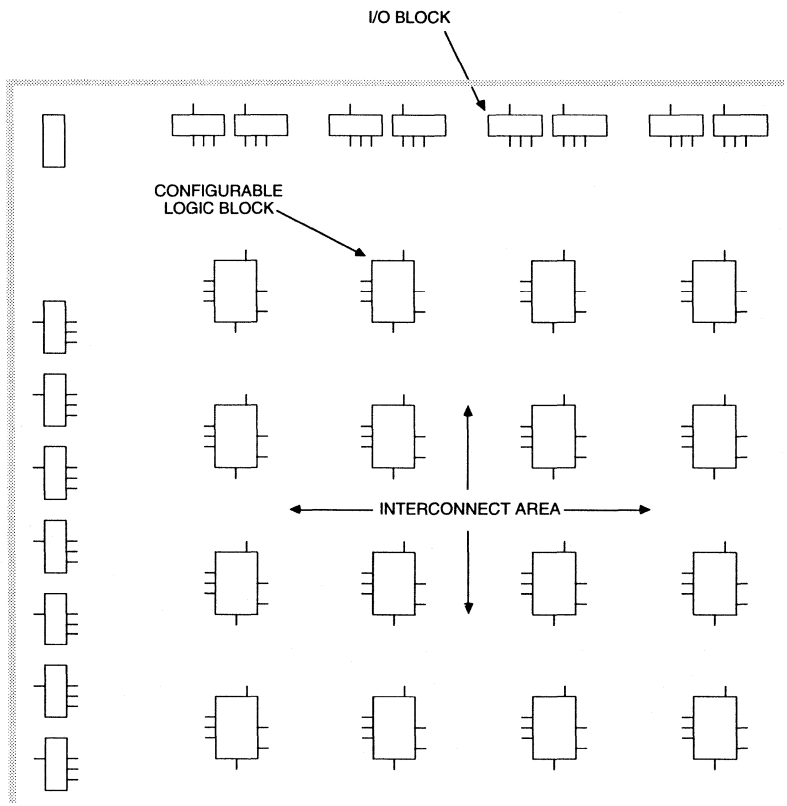
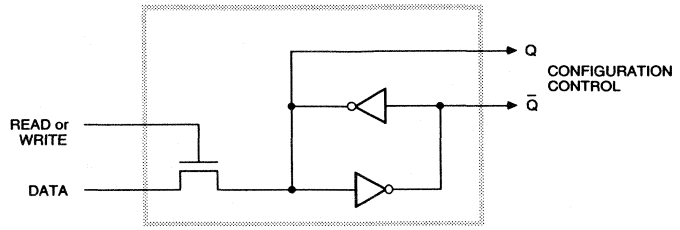


Figure 1. Logic Cell Array Structure

1104 01



1105 12

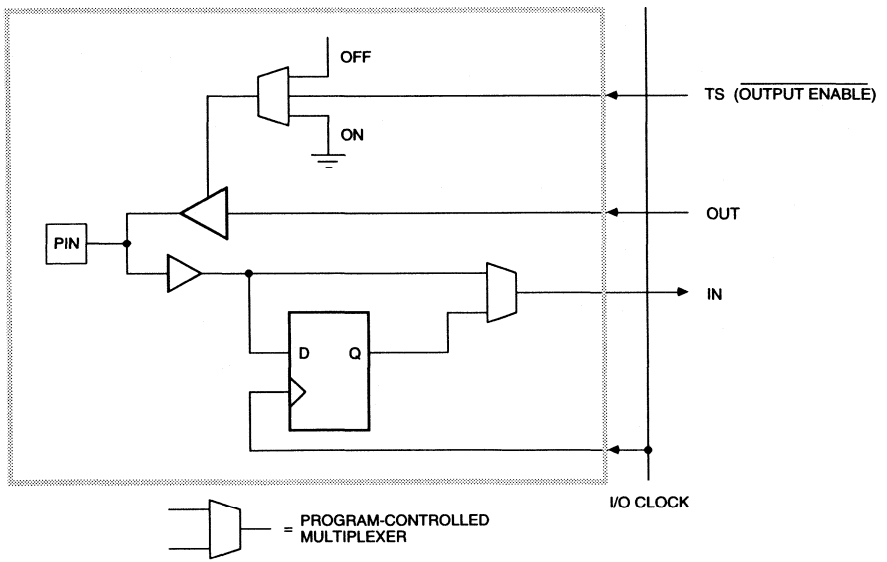
Figure 2. Configuration Memory Cell

edge-triggered D flip-flop and one input of a two-input multiplexer. The output of the flip-flop provides the other input to the multiplexer. The user can select either the direct input path or the registered input, based on the content of the memory cell controlling the multiplexer. The I/O Blocks along each edge of the die share common clocks. The flip-flops are reset during configuration as well as by the active-low chip $\overline{\text{RESET}}$ input.

Output buffers in the I/O blocks provide 4-mA drive for high fan-out CMOS or TTL-compatible signal levels. The output data (driving I/O block pin O) is the data source for the I/O block output buffer. Each I/O block output buffer is controlled by the contents of two configuration memory cells which turn the buffer ON or OFF or select 3-state buffer control. The user may also select the output buffer 3-state control (I/O block pin TS). When this I/O block output control signal is High (a logic one), the buffer is disabled and the package pin is high-impedance.

Configurable Logic Block

An array of Configurable Logic Blocks (CLBs) provides the functional elements from which the user's logic is constructed. The logic blocks are arranged in a matrix in the



1104 03

Figure 3. I/O Block

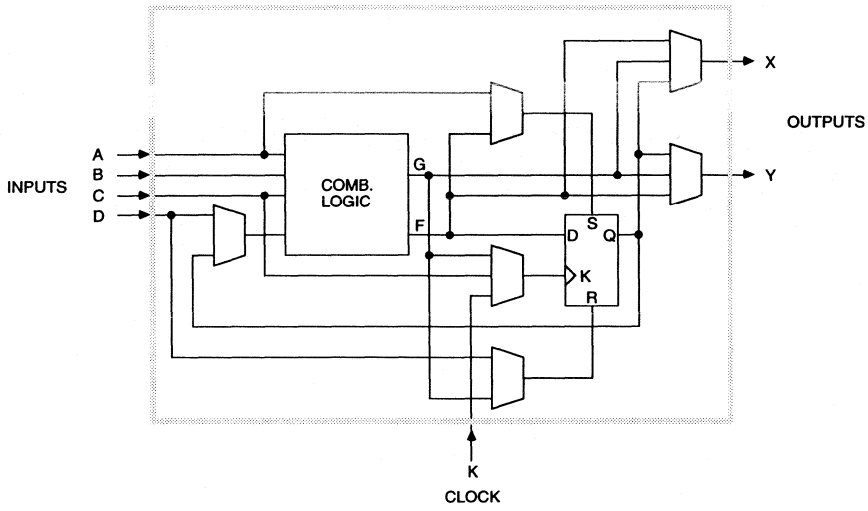


Figure 4. Configurable Logic Block

center of the device. The XC2064 has 64 such blocks arranged in an 8-row by 8-column matrix. The XC2018 has 100 logic blocks arranged in a 10 by 10 matrix.

Each logic block has a combinatorial logic section, a storage element, and an internal routing and control section. Each CLB has four general-purpose inputs: A, B, C and D; and a special clock input (K), which may be driven from the interconnect adjacent to the block. Each CLB also has two outputs, X and Y, which may drive interconnect networks. Figure 4 shows the resources of a Configurable Logic Block.

The logic block combinatorial logic uses a table look-up memory to implement Boolean functions. This technique can generate any logic function of up to four variables with a high speed sixteen-bit memory. The propagation delay through the combinatorial network is independent of the function generated. Each block can perform any function of four variables or any two functions of three variables each. The variables may be selected from among the four inputs and the block's storage element output Q. Figure 5 shows various options which may be specified for the combinatorial logic.

If the single 4-variable configuration is selected (Option 1), the F and G outputs are identical. If the 2-function alternative is selected (Option 2), logic functions F and G may be independent functions of three variables each. The three variables can be selected from among the four logic block inputs and the storage element output Q. A third form of the combinatorial logic (Option 3) is a special case of the 2-function form in which the B input dynamically selects between the two function tables providing a single merged logic function output. This dynamic selection allows some 5-variable functions to be generated from the four block inputs and storage element Q. Combinatorial functions are restricted in that one may not use both its storage element output Q and the input variable of the logic block pin "D" in the same function.

If used, the storage element in each Configurable Logic Block (Figure 6) can be programmed to be either an edge-sensitive "D" type flip-flop or a level-sensitive "D" latch. The clock or enable for each storage element can be selected from:

- The special-purpose clock input K
- The general-purpose input C
- The combinatorial function G

The user may also select the clock active sense within each logic block. This programmable inversion elimi-nates the need to route both phases of a clock signal throughout the device.

The storage element data input is supplied from the function F output of the combinatorial logic. Asynchronous SET and RESET controls are provided for each storage element. The user may enable these controls independently and select their source. They are active High inputs and the asynchronous reset is dominant. The

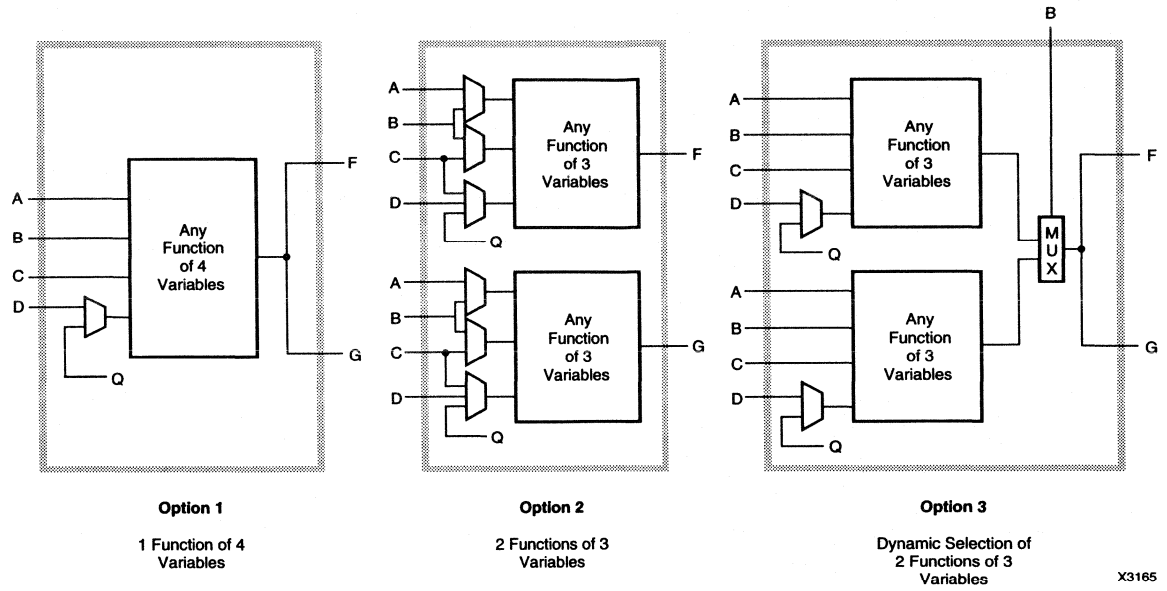


Figure 5. CLB Combinatorial Logic Options
Note: Variables D and Q can not be used in the same function.

storage elements are reset by the active-Low chip $\overline{\text{RESET}}$ pin as well as by the initialization phase preceding configuration. If the storage element is not used, it is disabled.

The two block outputs, X and Y, can be driven by either the combinatorial functions, F or G, or the storage element output Q (Figure 4). Selection of the outputs is completely interchangeable and may be made to optimize routing efficiencies of the networks interconnecting the logic blocks and I/O blocks.

Programmable Interconnect

Programmable interconnection resources in the Logic Cell Array provide routing paths to connect inputs and outputs of the I/O and logic blocks into desired networks. All interconnections are composed of metal segments, with programmable switching points provided to implement the necessary routing. Three types of resources accommodate different types of networks.

- General purpose interconnect
- Longlines
- Direct connection

General-Purpose Interconnect

General-purpose interconnect, as shown in Figure 7a, is composed of four horizontal metal segments between the rows and five vertical metal segments between the columns of logic and I/O blocks. Each segment is only the height or width of a logic block. Where these segments would cross at the intersections of rows and columns, switching matrices are provided to allow interconnections of metal segments from the adjoining rows and columns. Switches in the switch matrices and on block outputs are specially designed transistors, each controlled by a configuration bit.

Logic-block output switches provide contacts to adjacent general interconnect segments and therefore to the switching matrix at each end of those segments. A switch matrix

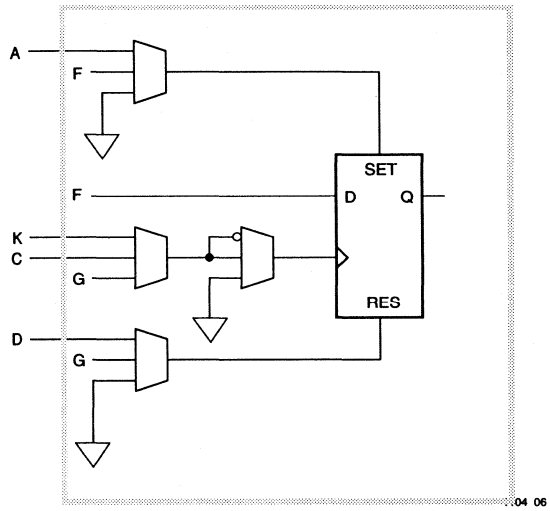


Figure 6. CLB Storage Elememt

can connect an interconnect segment to other segments to form a network. Figure 7a shows the general interconnect used to route a signal from one logic block to three other logic blocks. As shown, combinations of closed switches in a switch matrix allow multiple branches for each network. The inputs of the logic or I/O blocks are multiplexers that can be programmed with configuration bits to select an input network from the adjacent interconnect segments. Since the switch connections to block inputs are unidirectional (as are block outputs) they are usable *only* for input connection. The development system software provides automatic routing of these interconnections. Interactive routing is also available for design optimization. This is accomplished by selecting a networkand then toggling the states of the interconnect points by selecting them with the "mouse". In this mode, the connections through the switch matrix may be established by selecting pairs of matrix pins. The switching matrix combinations are indicated in Figure 7b.

Special buffers within the interconnect area provide periodic signal isolation and restoration for higher general

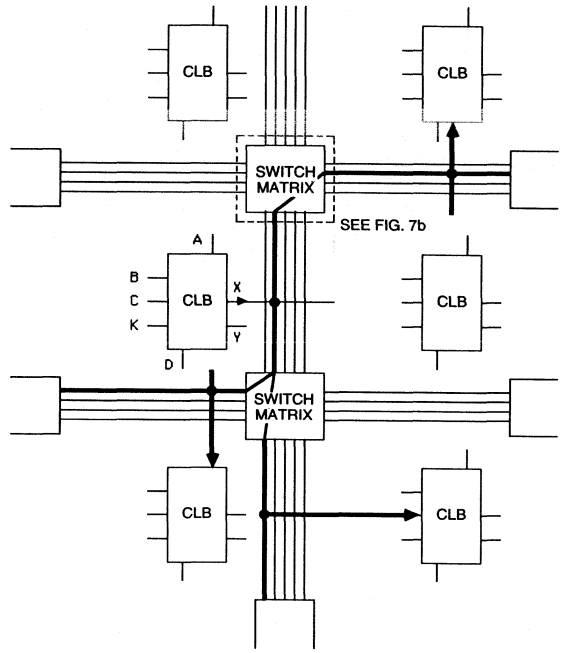


Figure 7a. General-Purpose Interconnect

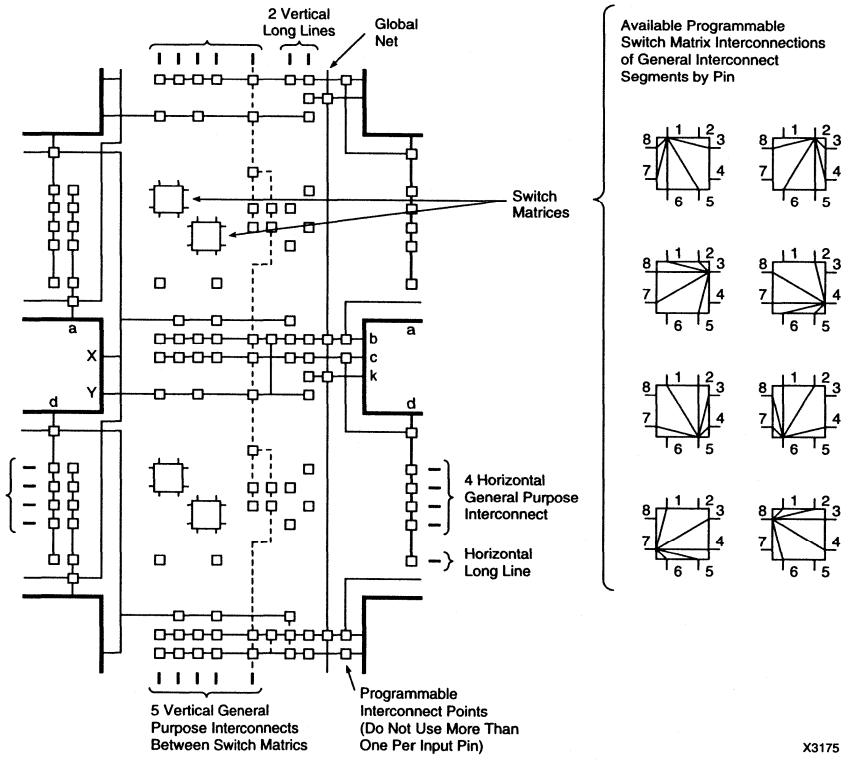


Figure 7b. Routing and Switch Matrix Connections

interconnect fan-out and better performance. The repowering buffers are bidirectional, since signals must be able to propagate in either direction on a general interconnect segment. Direction controls are automatically established by the Logic Cell Array development system software. Repowering buffers are provided only for the general-purpose interconnect since the direct and Longline resources do not exhibit the same R-C delay accumulation. The Logic Cell Array is divided into nine sections with buffers automatically provided for general interconnect at the boundaries of these sections. These boundaries can be viewed with the development system. For routing within a section, no buffers are used. The delay calculator of the XACT development system automatically calculates and displays the block, interconnect and buffer delays for any selected paths.

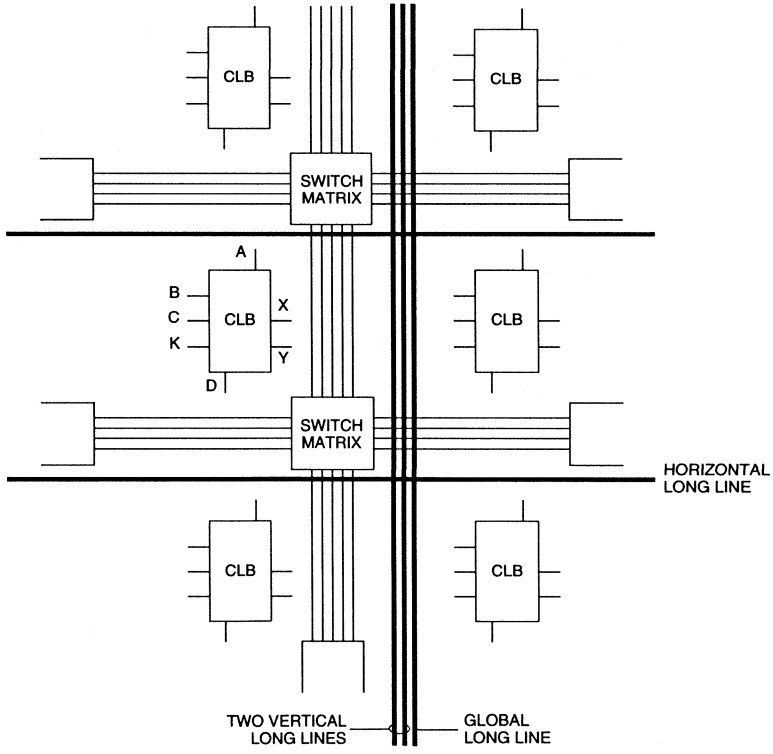
Longlines

Longlines, shown in Figure 8a, run both vertically and horizontally the height or width of the interconnect area. Each vertical interconnection column has two Longlines; each horizontal row has one, with an additional Longline adjacent to each set of I/O blocks. The Longlines bypass the switch matrices and are intended primarily for signals that must travel a long distance or must have minimum skew among multiple destinations.

A global buffer in the Logic Cell Array is available to drive a single signal to all B and K inputs of logic blocks. Using the global buffer for a clock provides a very low skew, high fan-out synchronized clock for use at any or all of the logic blocks. At each block, a configuration bit for the K input to the block can select this global line as the storage element clock signal. Alternatively, other clock sources can be used.

A second buffer below the bottom row of the array drives a horizontal Longline which, in turn, can drive a vertical Longline in each interconnection column. This alternate buffer also has low skew and high fan-out capability. The network formed by this alternate buffer's Longlines can be selected to drive the B, C or K inputs of the logic blocks.

Alternatively, these Longlines can be driven by a logic or I/O block on a column by column basis. This capability provides a common, low-skew clock or control line within each column of logic blocks. Interconnections of these Longlines are shown in Figure 8b.



1104 09

Figure 8a. Longline Interconnect

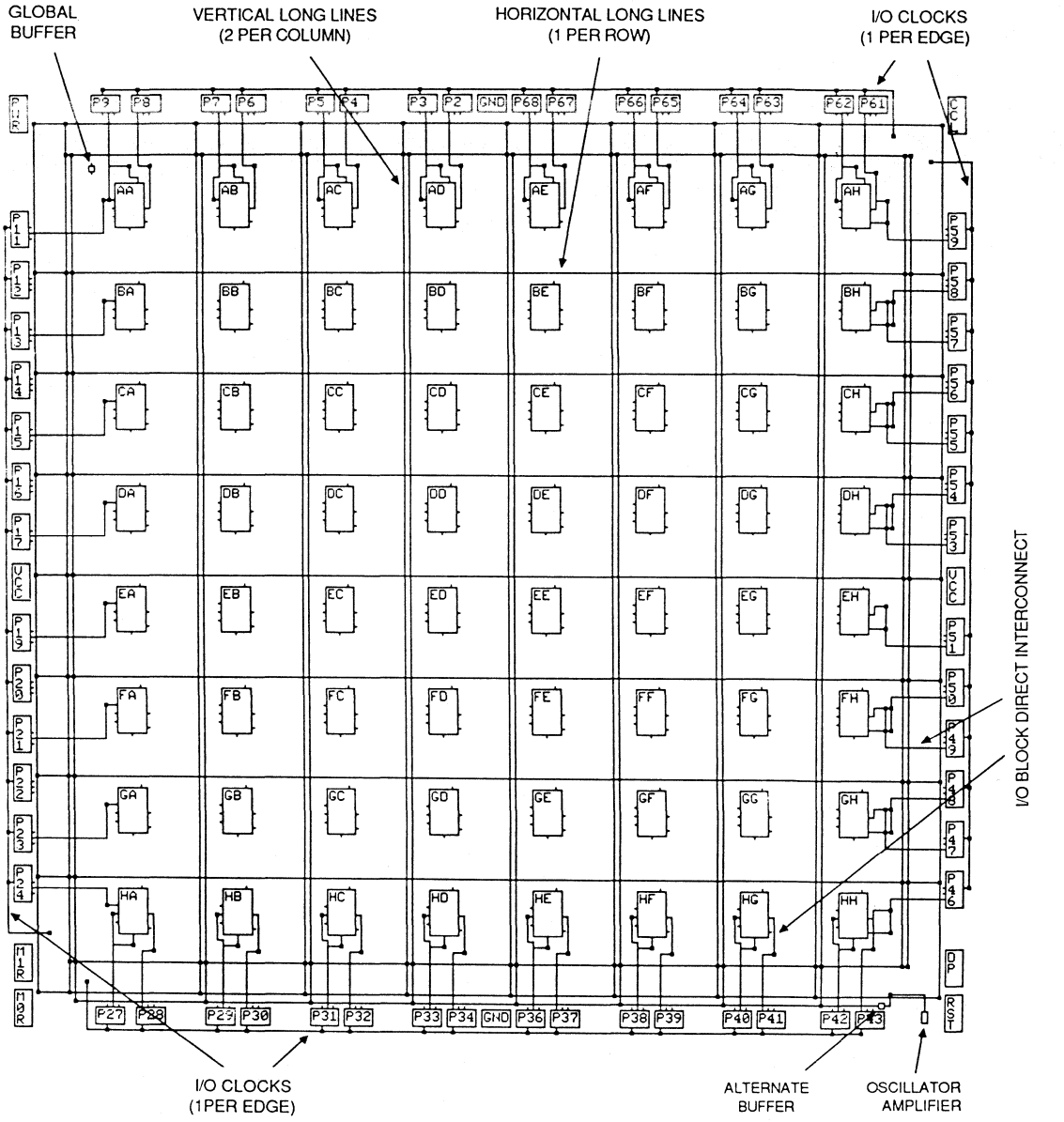


Figure 8b. XC2064 Longlines, I/O Clocks, I/O Direct Interconnect

Direct Interconnect

Direct interconnect, shown in Figure 9, provides the most efficient implementation of networks between adjacent logic or I/O blocks. Signals routed from block to block by means of direct interconnect exhibit minimum interconnect propagation and use minimum interconnect resources. For each Configurable Logic Block, the X output may be connected directly to the C or D inputs of the CLB above and to the A or B inputs of the CLB below it. The Y output can use direct interconnect to drive the B input of the block immediately to its right. Where logic blocks are adjacent to I/O blocks, direct connect is provided to the I/O block input (I) on the left edge of the die, the output (O) on the right edge, or both on I/O blocks at the top and bottom of the die. Direct interconnections of I/O blocks with CLBs are shown in Figure 8b.

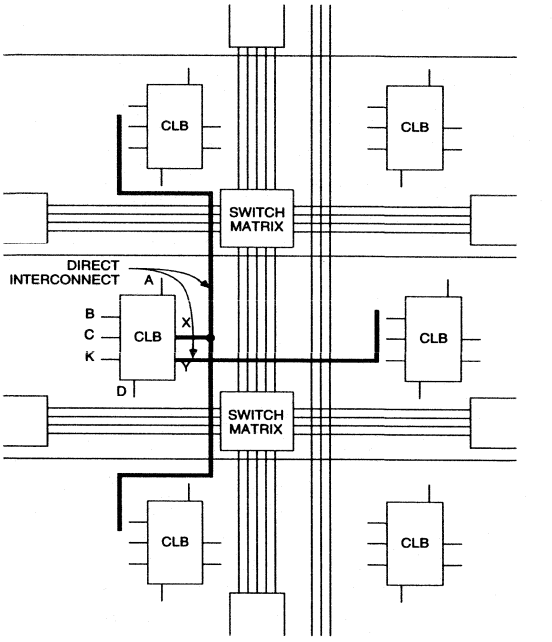
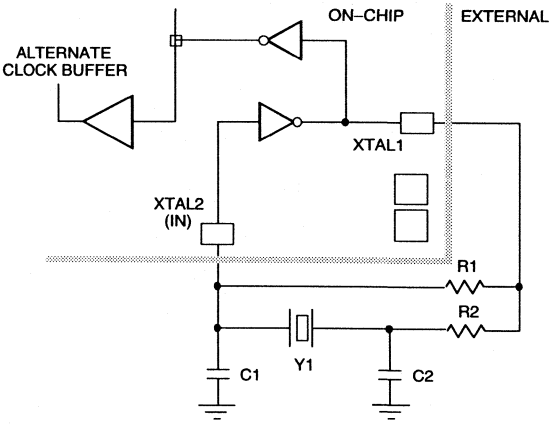


Figure 9. Direct Interconnect

Crystal Oscillator

Figure 8b also shows the location of an internal high speed inverting amplifier which may be used to implement an on-chip crystal oscillator. It is associated with the auxiliary buffer in the lower right corner of the die. When the oscillator is configured by MAKEBITS and connected as a signal source, two special user IOBs are also configured to connect the oscillator amplifier with external crystal oscillator components as shown in Figure10. The oscillator circuit becomes active early in the configuration process in order to allow the oscillator to stabilize. Actual internal connection is delayed until completion of configuration. In Figure 10, the feedback resistor R1, between the output and input, biases the amplifier at threshold. The inversion of the amplifier, together with the R-C networks and an AT-cut series resonant crystal, produce the 360-degree phase shift of the Pierce oscillator. A series resistor R2 may be included to add to the amplifier output impedance when needed for phase-shift control, crystal resistance matching, or to limit the amplifier input swing to control clipping at large amplitudes. Excess feedback voltage may be corrected by the ratio of C2/C1. The amplifier is designed to be used from 1 MHz to about one-half the specified CLB toggle frequency. Use at frequencies below 1 MHz may require individual characterization with respect to a series resistance. Crystal oscillators above 20 MHz generally require a crystal which operates in a third overtone mode, where the fundamental frequency must be suppressed by an inductor across C2, turning this parallel resonant circuit to double the fundamental crystal frequency, i.e., 2/3 of the desired third harmonic frequency network. When the oscillator inverter is not used, these IOBs and their package pins are available for general user I/O.



SUGGESTED COMPONENT VALUES
R1 0.5 – 1 MΩ
R2 0 – 1 KΩ (may be required for low frequency, phase shift and/or compensation level for crystal Q)
C1, C2 10 – 40 pF
Y1 1 – 20 MHz AT cut series resonant

	XTAL1	XTAL2
48 DIP	33	30
68 PLCC	46	43
68 PGA	J10	L10
84 PLCC	56	53
84 PGA	K11	L11

Figure 10. Crystal Oscillator

Programming

Table 1. Configuration Mode Selection

M0	M1	M2	CCLK	Mode	Data
0	0	0	output	Master	Bit Serial
0	0	1	output	Master	Byte Wide Addr. = 0000 up
0	1	0	—	reserved	—
0	1	1	output	Master	Byte Wide Addr. = FFFF down
1	0	0	—	reserved	—
1	0	1	output	Peripheral	Bit Serial
1	1	0	—	reserved	—
1	1	1	input	Slave	Bit Serial

Configuration data to define the function and interconnection within a Logic Cell Array are loaded automatically at power-up or upon command. Several methods of automatically loading the required data are designed into the Logic Cell Array and are determined by logic levels applied to mode selection pins at configuration time. The form of the data may be either serial or parallel, depending on the configuration mode. The programming data are independent of the configuration mode selected. The state diagram of Figure 11 illustrates the configuration process.
Input thresholds for user I/O pins can be selected to be either TTL-compatible or CMOS-compatible. At power-up, all inputs are TTL-compatible and remain in that state until the LCA begins operation. If the user has selected CMOS compatibility, the input thresholds are changed to CMOS levels during configuration.

Figure 12 shows the specific data arrangement for the XC2064 device. Future products will use the same data format to maintain compatibility between different devices of the Xilinx product line, but they will have different sizes and numbers of data frames. For the XC2064, configuration requires 12,038 bits for each device. For the XC2018, the configuration of each device requires 17,878 bits. The XC2064 uses 160 configuration data frames and the XC2018 uses 197.

The configuration bit stream begins with preamble bits, a preamble code and a length count. The length count is loaded into the control logic of the Logic Cell Array and is used to determine the completion of the configuration process. When configuration is initiated, a 24-bit length counter is set to 0 and begins to count the total number of configuration clock cycles applied to the device. When the current length count equals the loaded length count, the configuration process is complete. Two clocks before completion, the internal logic becomes active and is reset. On the next clock, the inputs and outputs become active as configured and consideration should be given to avoid configuration signal contention. *(Attention must be paid to avoid contention on pins which are used as inputs during configuration and become outputs in operation.)* On the last configuration clock, the completion of configuration is signalled by the release of the DONE / $\overline{\text{PROG}}$ pin of the device as the device begins operation. This open-drain output can be AND-tied with multiple Logic Cell Arrays and used as an active-High READY or active-Low , $\overline{\text{RESET}}$, to other portions of the system. High during configuration (HDC) and low during configuration ($\overline{\text{LDC}}$), are released one CCLK cycle before DONE is asserted. In master mode configurations, it is convenient to use $\overline{\text{LDC}}$ as an active-Low EPROM chip enable.

As each data bit is supplied to the LCA, it is internally assembled into a data word. As each data word is completely assembled, it is loaded in parallel into one word of the internal configuration memory array. The last word must be loaded before the current length count compare is true. If the configuration data are in error, e.g., PROM address lines swapped, the LCA will not be ready at the length count and the counter will cycle through an additional complete count prior to configuration being "done".

Table 1 shows the selection of the configuration mode based on the state of the mode pins M0 and M1. These package pins are sampled prior to the start of the configuration process to determine the mode to be used. Once configuration is DONE and subsequent operation has begun, the mode pins may be used to perform data readback, as discussed later. An additional mode pin, M2, must be defined at the start of configuration. This package pin is a user-configurable I/O after configuration is complete.

Initialization Phase
When power is applied, an internal power-on-reset circuit is triggered. When Vcc reaches the voltage at which the LCA device begins to operate (nominally 2.5 to 3 V), the chip is initialized, outputs are made high-impedance and a time-out is initiated to allow time for power to stabilize. This time-out (11 to 33 ms) is determined by a counter driven by a self-generated, internal sampling clock that drives the configuration clock (CCLK) in master configuration mode. This internal sampling clock will vary with process, temperature and power supply over the range of 0.5 to 1.5 MHz. LCA devices with mode lines set for master mode will time-out of their initialization using a longer counter (43 to 130 ms) to assure that all devices, which it may be driving in a daisy chain, will be ready. Configuration using peripheral or slave modes must be delayed long enough for this initialization to be completed.

The initialization phase may be extended by asserting the active-Low external $\overline{\text{RESET}}$. If a configuration has begun, an assertion of $\overline{\text{RESET}}$ will initiate an abort, including an orderly clearing of partially loaded configuration memory bits. After about three clock cycles for synchronization, initialization will require about 160 additional cycles of the internal sampling clock (197 for the XC2018) to clear the internal memory before another configuration may begin.

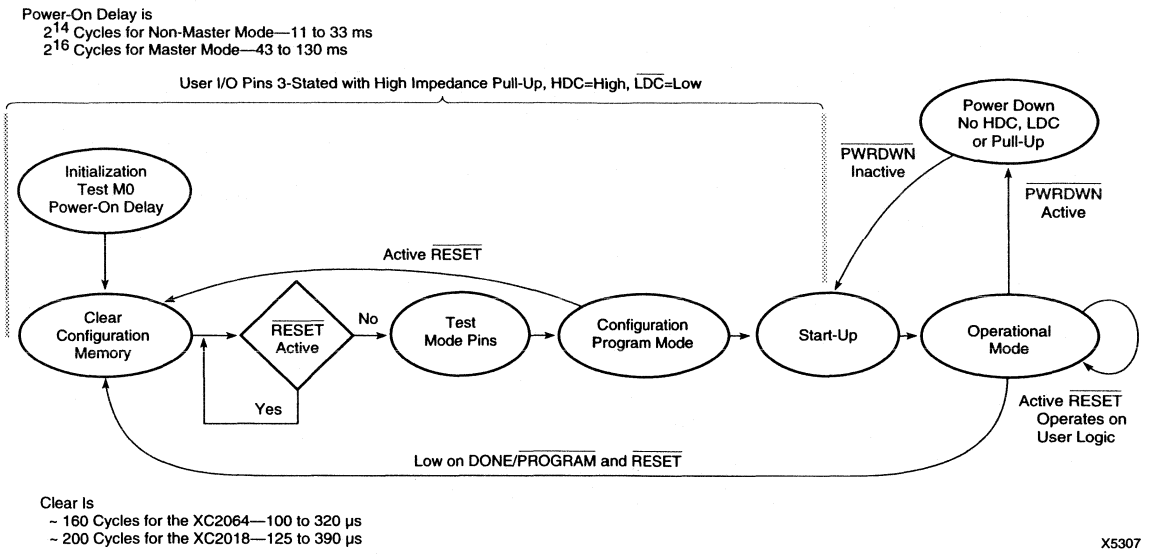


Figure 11. A State Diagram of the Configuration Process for Power-up and Reprogram

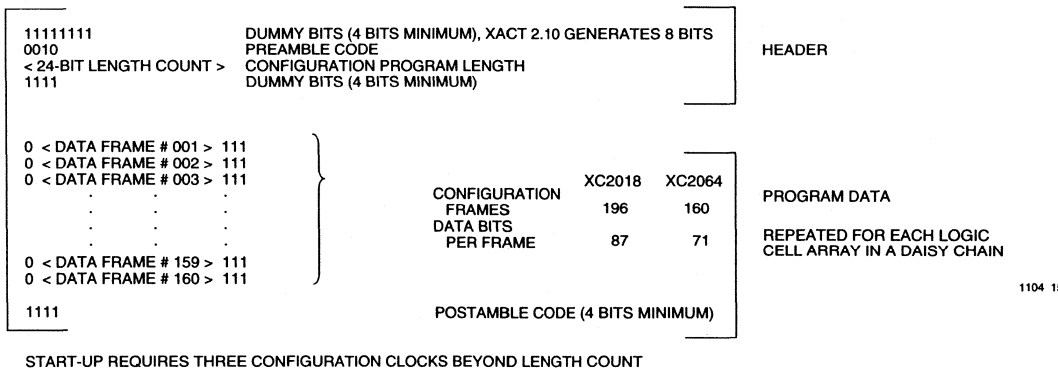


Figure 12. XC2064 Internal Configuration Data Arrangement

Reprogramming is initialized by a High-to-Low transition on $\overline{\text{RESET}}$ (after $\overline{\text{RESET}}$ has been High for at least 6 µs) followed by a Low level (for at least 6 µs) on both the $\overline{\text{RESET}}$ and the open-drain DONE/$\overline{\text{PROG}}$ pins. This returns the LCA device to the CLEAR state, as shown in Figure 11.

Master Mode

In Master mode, the Logic Cell Array automatically loads the configuration program from an external memory device. The Master Serial mode uses serial configuration data, synchronized by the rising edge of CCLK, as shown in Figure 13.

In Master Parallel mode (Figure 14), the Logic Cell Array provides 16 address outputs and the control signals RCLK (Read Clock), HDC (High during configuration) and LDC (Low during configuration) to execute Read cycles from the external memory. Parallel 8-bit data words are read and internally serialized. As each data word is read, the least significant bit of each byte, normally D0, is the next bit in the serial stream.

Addresses supplied by the Logic Cell Array can be selected by the mode lines to begin at address 0 and incremented to reach the memory (master Low mode), or they can begin at address FFFF Hex and be decremented

(master High mode). This capability is provided to allow the Logic Cell Array to share external memory with another device, such as a microprocessor. For example, if the processor begins its execution from Low memory, the Logic Cell Array can load itself from High memory and enable the processor to begin execution once configuration is completed. The Done/$\overline{PROG}$ output pin can be used to hold the processor in a Reset state until the Logic Cell Array has completed the configuration process

Peripheral Mode (Bit Serial)

Peripheral mode provides a simplified interface through which the device may be loaded as a processor peripheral. Figure 15 shows the peripheral mode connections. Processor Write cycles are decoded from the common assertion of the active-Low write strobe (IOWRT), and two active-Low and of the active-High chip selects (CS0 CS1 CS2). If all these signals are not available, the unused inputs should be driven to their respective active levels. The Logic Cell Array will accept one bit of the configuration program on the data input (DIN) pin for each processor Write cycle. Data is supplied in the serial sequence described earlier.

Since only a single bit from the processor data bus is loaded per cycle, the loading process involves the processor reading a byte or word of data, writing a bit of the data to the Logic cell Array, shifting the word and writing a bit until all bits of the word are written, then continuing in the same fashion with the next word, etc. After the configuration program has been loaded, an additional three clocks (a total of three more than the length count) must be supplied in order to complete the configuration process. When more than one device is being used in the system, each device can be assigned a different bit in the processor data bus, and multiple devices can be loaded on each processor write cycle. This broadside loading method provides a very easy and time-efficient method of loading several devices.

Slave Mode

Slave mode, Figure 16, provides the simplest interface for loading the Logic Cell Array configuration. Data is supplied in conjunction with a synchronizing clock. For each Low-to-High input transition of configuration clock (CCLK), the data present on the data input (DIN) pin is loaded into the internal shift register. Data may be supplied by a processor or by other special circuits. Slave mode is used for downstream devices in a daisy-chain configuration. The data for each slave LCA device are supplied by the preceding LCA device in the chain, and the clock is supplied by the lead device, which is configured in master or peripheral mode. After the configuration program has been loaded, an additional three clocks (a total of three more than the length count) must be supplied in order to complete the configuration process.

Daisy Chain

The daisy-chain programming mode is supported by Logic Cell Arrays in all programming modes. In master mode and peripheral modes, the LCA device can act as a source of data and control for slave devices. For example, Figure 14 shows a single device in master mode, with two devices in slave mode. The master-mode device reads the external memory and begins the configuration loading process for all of the devices.

The data begins with a preamble and a length count which are supplied to all devices at the beginning of the configuration. The length count represents the total number of cycles required to load all of the devices in the daisy chain. After loading the length count, the lead device will load its configuration data while providing a High DOUT to downstream devices. When the lead device has been loaded and the current length count has not reached the full value, memory access continues. Data bytes are read and serialized by the lead device. The data is passed through the lead device and appears on the data out (DOUT) pin in serial form. The lead device also generates the configuration clock (CCLK) to synchronize the serial output data. A master-mode device generates an internal CCLK of eight times the EPROM address rate, while a peripheral mode device produces CCLK from the chip select and write strobe timing.

Operation

When all of the devices have been loaded and the length count is complete, a synchronous start-up of operation is performed. On the clock cycle following the end of loading, the internal logic begins functioning in the reset state. On the next CCLK, the configured output buffers become active to allow signals to stabilize. The next CCLK cycle produces the DONE condition. The length count control of operation allows a system of multiple Logic Cell Arrays to begin operation in a synchronized fashion. If the crystal oscillator is used, it will begin operation before configuration is complete to allow time for stabilization before it is connected to the internal circuitry.

Reprogram

The Logic Cell Array configuration memory may be rewritten while the device is operating in the user's system. The LCA device returns to the Clear state where the configuration memory is cleared, I/O pins disabled, and mode lines re-sampled. Reprogram control is often implemented using an external open collector driver which pulls DONE/$\overline{PROG}$ LOW. Once it recognizes a stable request, the Logic Cell Array holds DONE/$\overline{PROG}$ LOW until the new configuration has been completed. Even if the DONE/$\overline{PROG}$ pin is externally held LOW beyond the configuration period, the Logic Cell Array begins operation upon completion of configuration. To reduce sensitivity to noise, these re-program signals are filtered for 2–3 cycles of the

LCA internal timing generator (2 to 6 µs). Note that the Clear time-out for a master-mode reprogram or abort does not have the 4 times delay of the Initialization state. If a daisy chain is used, an external $\overline{\text{RESET}}$ is required, long enough to guarantee clearing all non-master mode devices. For XC2000-series LCA devices, this is accomplished with an external time delay.

In some applications the system power supply might have momentary failures which can leave the LCA control logic in an invalid state. There are two methods to recover from this state. The first is to cycle the V_{CC} supply to less than 0.1 V and re-apply valid V_{CC}. The second is to provide the LCA device with simultaneous Low levels of at least 6 µs on $\overline{\text{RESET}}$ and DONE/$\overline{\text{PROG}}$ pins after the $\overline{\text{RESET}}$ pin has been High following a return to valid V_{CC}. This guarantees that the LCA device will return to the Clear state. Either of these methods may be needed in the event of an incomplete voltage interruption. They are not needed for a normal application of power from an off condition.

Battery Backup

Because the control store of the Logic Cell Array is a CMOS static memory, its cells require only a very low standby current for data retention. In some systems, this low data-retention current characteristic facilitates preserving configurations in the event of a primary power loss. The Logic Cell Array has built in power-down logic which, when activated, clears all internal flip-flops and latches, but retains the configuration. All outputs are placed in the high-impedance state, and all input levels are ignored. The internal logic considers all inputs to be ones (High). Configuration is not possible during power down.

Power-down data retention is possible with a simple battery-backup circuit because the power requirement is extremely low. For retention at 2.0 V, the required current is typically on the order of 500 nA. Screening to this parameter is available. To force the Logic Cell Array into the power-down state, the user must pull the $\overline{\text{PWRDWN}}$ pin Low and continue to supply a retention voltage to the V_{CC} pins of the package. When normal power is restored, V_{CC} is elevated to its normal operating voltage and $\overline{\text{PWRDWN}}$ is returned to a High. The Logic Cell Array resumes operation with the same internal sequence that occurs at the conclusion of configuration. Internal I/O and logic block storage elements will be reset, the outputs will become enabled and then the DONE/$\overline{\text{PROG}}$ pin will be released. No configuration programming is involved.

Special Configuration Functions

In addition to the normal user logic functions and interconnect, the configuration data include control for several special functions:

- Input thresholds
- Readback disable
- DONE pull-up resistor

Each of these functions is controlled by a portion of the configuration program generated by the XACT Development System.

Input Thresholds

During configuration, all input thresholds are TTL level. After configuration, input thresholds are established as specified, either TTL or CMOS. The $\overline{\text{PWRDWN}}$ input threshold is an exception; it is always a CMOS level input. The TTL threshold option requires additional power for threshold shifting.

Readback

After a Logic Cell Array has been programmed, the configuration program may be read back from the device. Readback may be used for verification of configuration, and as a method of determining the state of internal logic nodes during debugging. Three Readback options are provided: on command, only once, and never.

An initiation of Readback is produced by a Low-to-High transition of the M0 / RTRIG (Read Trigger) pin. The CCLK input must then be driven by external logic to read back the configuration data. The first three Low-to-High CCLK transitions clock out dummy data. The subsequent Low-to-High CCLK transitions shift the data frame information out on the M1/$\overline{\text{RDATA}}$ (Read Data) pin. Note that the logic polarity is always inverted, a zero in Configuration becomes a one in Readback, and vice versa. Note also that each Readback frame has one Start bit (read back as a one) but, unlike in Configuration, each Readback frame has only one Stop bit (read back as a zero). The third leading dummy bit mentioned above can be considered the Start bit of the first frame.

All data frames must be read back to complete the process and return the Mode Select and CCLK pins to their normal functions. Readback data includes the state of all internal storage elements. This information is used by the XACT development system In-Circuit Debugger to provide visibility into the internal operation of the logic while the system is operating. To read back a uniform time sample of all storage elements, it may be necessary to inhibit the system clock.

DONE Pull-up

The DONE / $\overline{\text{PROG}}$ pin is an open drain I/O that indicates programming status. As an input, it initiates a reprogram operation. An optional internal pull-up resistor may be enabled.

The following seven pages describe the four configuration modes in detail.

Master Serial Mode

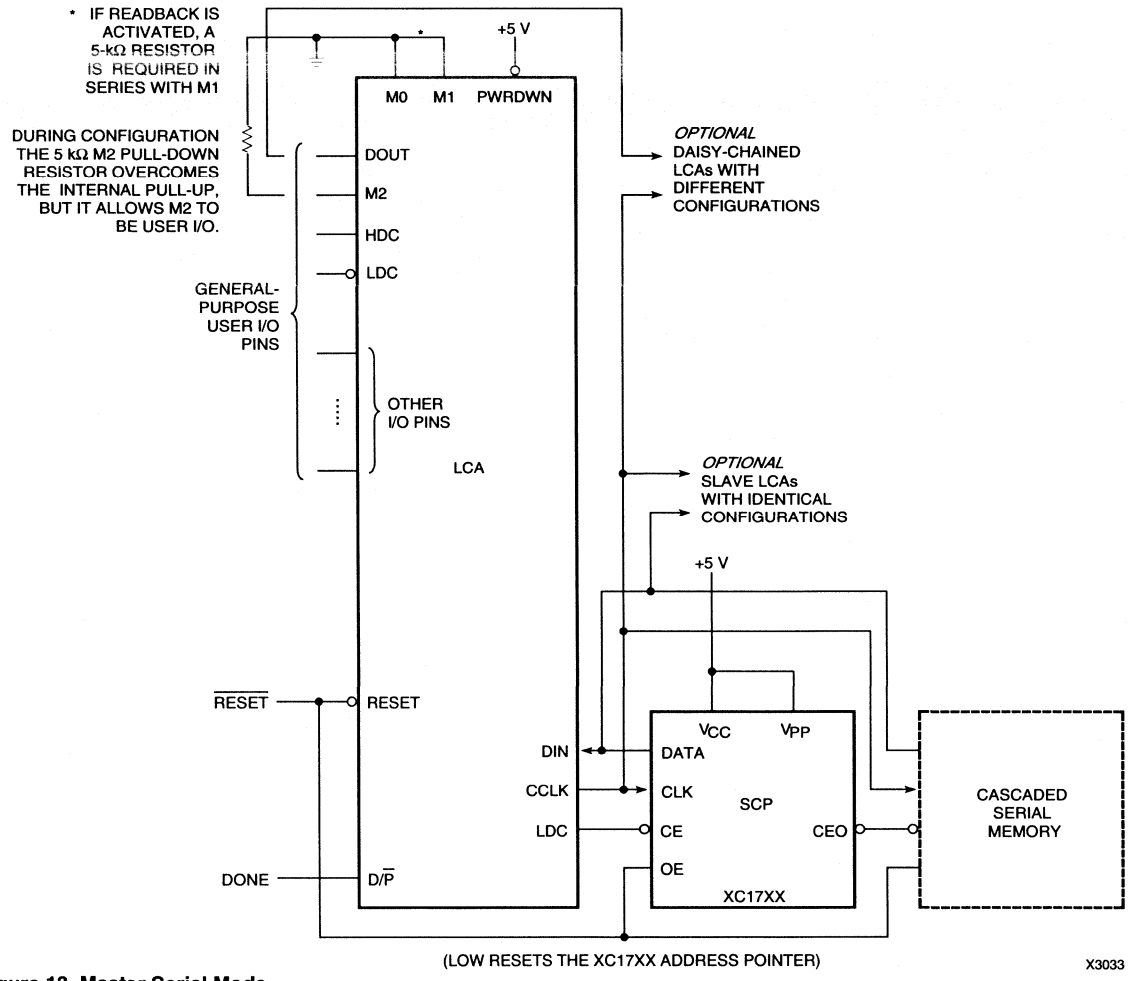


Figure 13. Master Serial Mode

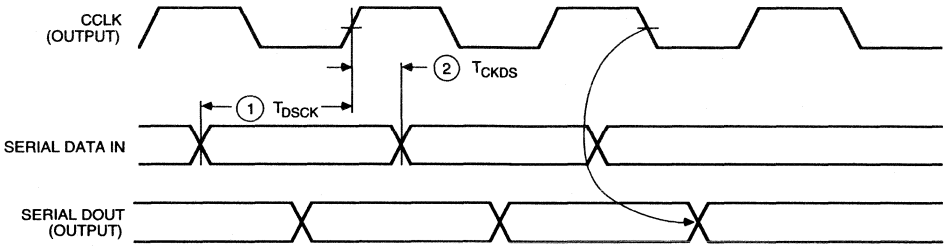
In Master Serial mode, the CCLK output of the lead LCA device drives a Xilinx Serial PROM that feeds the LCA DIN input. Each rising edge of the CCLK output increments the Serial PROM internal address counter. This puts the next data bit on the SPROM data output, connected to the LCA DIN pin. The lead LCA device accepts this data on the subsequent rising CCLK edge.

The lead LCA device then presents the preamble data (and all data that overflows the lead device) on its DOUT pin.

There is an internal delay of 1.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

The SPROM CE input can be driven from either $\overline{LDC}$ or DONE. Using $\overline{LDC}$ avoids potential contention on the DIN pin, if this pin is configured as user-I/O, but $\overline{LDC}$ is then restricted to be a permanently High user output. Using DONE also avoids contention on DIN, provided the "early DONE" option is invoked

Master Serial Mode Programming Switching Characteristics



1105 29

Speed Grade				-50		-70		-100		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
CCLK[2]	Data In setup	1	T_{DSCK}	60		60		60		ns
	Data In hold	2	T_{CKDS}	0		0		0		ns

Notes: 1. At power-up, V_{CC} must rise from 2.0 V to Vcc min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until V_{CC} has reached 4.0 V (2.5 V for XC2000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require a >1-µs High level on $\overline{RESET}$, followed by a >6-µs Low level on $\overline{RESET}$ and D/$\overline{P}$ after V_{CC} has reached 4.0 V (2.5 V for XC2000L).

2. Master-serial-mode timing is based on slave-mode testing.

Master Parallel Mode

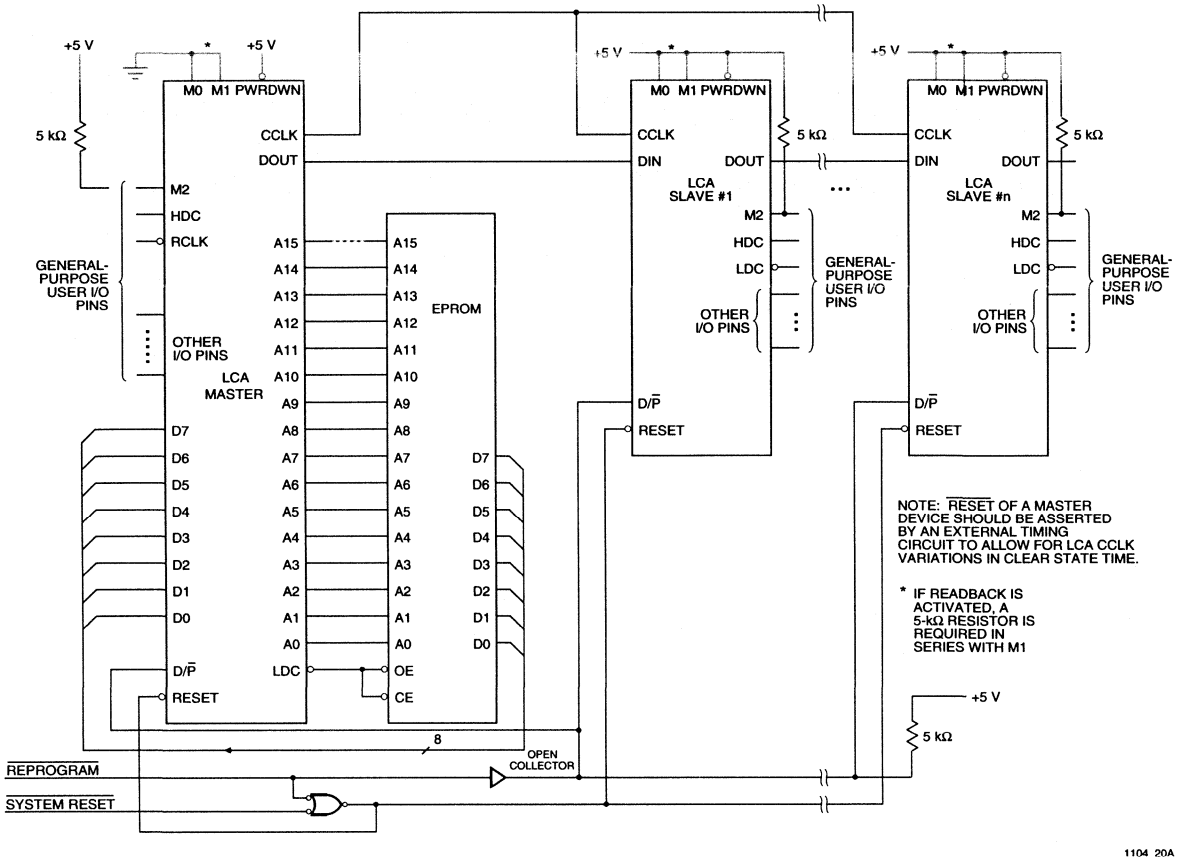


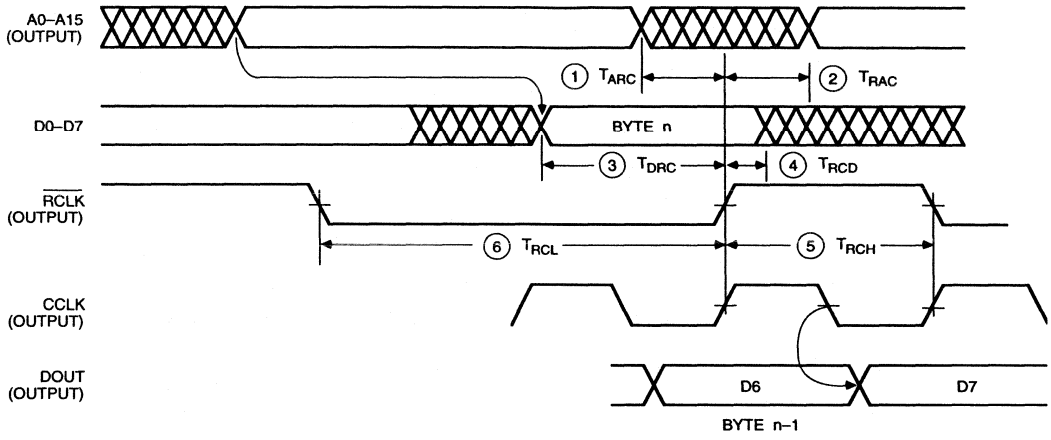
Figure 14. Master Parallel Mode Configuration with Daisy Chained Slave Mode Devices. All are configured from the common EPROM source. A well defined termination of SYSTEM RESET is needed when controlling multiple LCA devices.

In Master Parallel mode, the lead LCA device directly addresses an industry-standard byte-wide EPROM and accepts eight data bits right before incrementing (or decrementing) the address outputs.

The eight data bits are serialized in the lead LCA device, which then presents the preamble data (and all data that overflows the lead device) on the DOUT pin. There is an internal delay of 1.5 CCLK periods, after the rising CCLK edge that accepts a byte of data, and also changes the EPROM address, until the falling CCLK edge that makes the LSB (D0) of this byte appear at DOUT. This means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

Any XC3000 slave driven by an XC2000 master mode device must use early DONE and early internal reset. (The XC2000 master will not supply the extra clock required by a late programmed XC3000.)

Master Parallel Mode Programming Switching Characteristics



1104 33

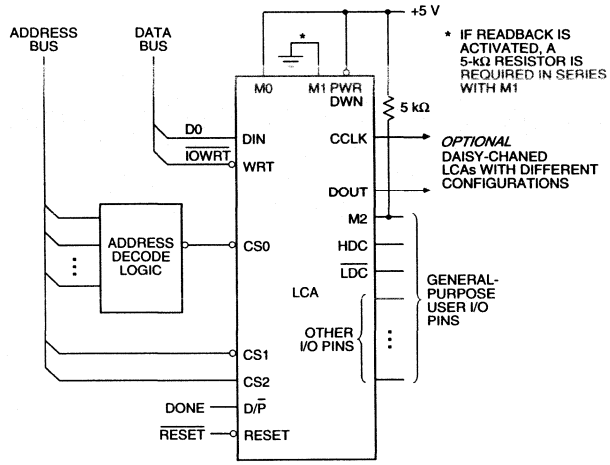
	Description	Symbol		Min	Max	Units
RCLK	From address invalid	1	T_{ARC}		0	ns
	To address valid	2	T_{RAC}		200	ns
	To data setup	3	T_{DRC}	60		ns
	To data hold	4	T_{RCD}	0		ns
	RCLK high	5	T_{RCH}	600		ns
	RCLK low	6	T_{RCL}	4.0		µs

Note: 1. CCLK and DOUT timing are the same as for slave mode.

2. At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until V_{CC} has reached 4.0 V (2.5 V for XC2000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require a >1-µs High level on RESET, followed by a >6-µs Low level on RESET and D/P after V_{CC} has reached 4.0 V (2.5 V for XC2000L).

This timing diagram shows that the EPROM requirements are extremely relaxed: EPROM access time can be longer than 4000 ns, EPROM data output has no hold time requirement

Peripheral Mode

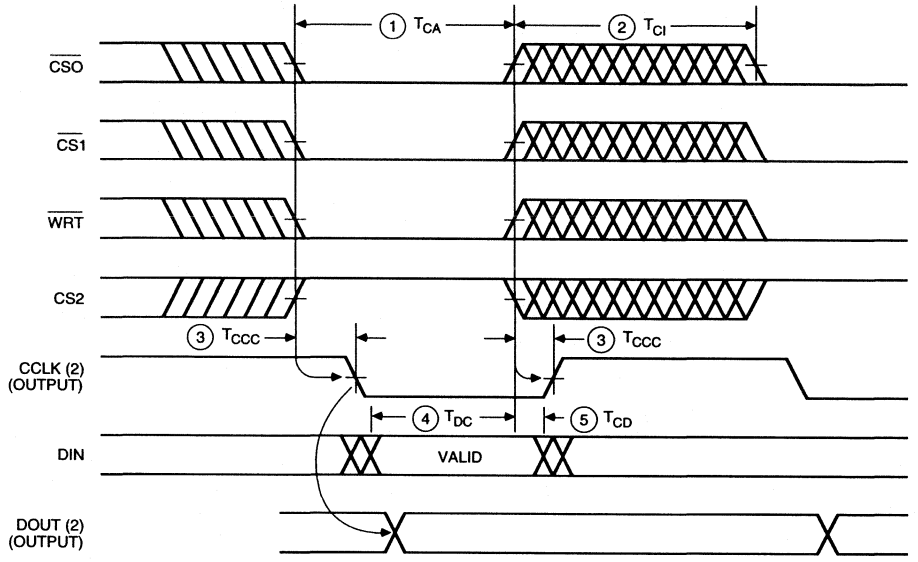


1104 18A

Figure 15. Peripheral Mode. Configuration data is loaded using serialized data from a microprocessor.

Peripheral mode uses the trailing edge of the logic AND condition of the $\overline{CS0}$, $\overline{CS1}$, CS2, and $\overline{WRT}$ inputs to accept bit-serial data from a microprocessor bus. In the lead LCA device, this data is serially shifted into the internal logic. The lead LCA device presents the preamble data (and all data that overflows the lead device) on the DOUT pin.

Peripheral Mode Programming Switching Characteristics



1104 34

	Description	Symbol		Min	Max	Units
Controls[1] ($\overline{CS0}$, CS1, $\overline{CS2}$, $\overline{WRT}$)	Active (last active input to first inactive)	1	T_{CA}	0.25	5.0	µs
	Inactive (first inactive input to last active)	2	T_{CI}	0.25		µs
	CCLK[2]	3	T_{CCC}		75	ns
	DIN setup	4	T_{DC}	50		ns
	DIN hold	5	T_{CD}	0		ns

Notes: 1. Peripheral mode timing determined from last control signal of the logical AND of (CS0, CS1, CS2, WRT) to transition to active or inactive state.

2. CCLK and DOUT timing are the same as for slave mode.

3. At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding $\overline{RESET}$ Low until Vcc has reached 4.0 V (2.5 V for XC2000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require a >1-µs High level on $\overline{RESET}$, followed by a >6-µs Low level on $\overline{RESET}$ and D/P after V_{CC} has reached 4.0 V (2.5 V for XC2000L).

Slave Serial Mode

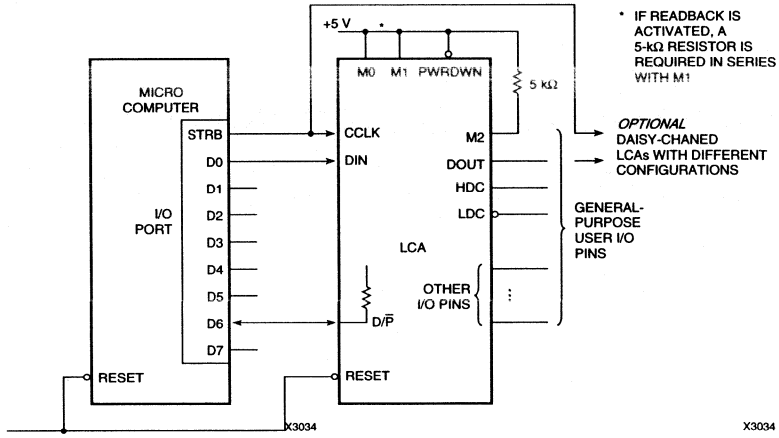
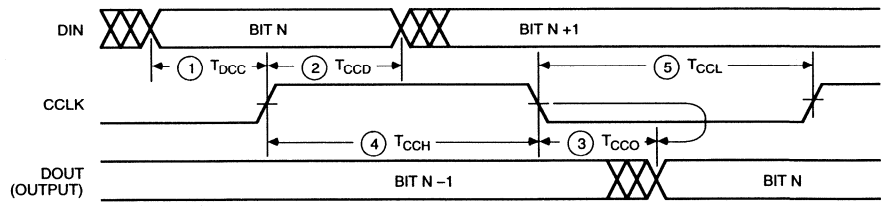


Figure 16. Slave Serial Mode. Bit-serial configuration data are read at rising edge of the CCLK. Data on DOUT are provided on the falling edge of CCLK. Identically configured non-master mode LCAs can be configured in parallel by connecting DINs and CCLKs.

In Slave Serial mode, an external signal drives the CCLK input(s) of the LCA device(s). The serial configuration bitstream must be available at the DIN input of the lead LCA device a short set-up time before each rising CCLK edge. The lead LCA device then presents the preamble data (and all data that overflows the lead device) on its DOUT pin.

There is an internal delay of 1.5 CCLK periods, which means that DOUT changes on the falling CCLK edge, and the next LCA device in the daisy-chain accepts data on the subsequent rising CCLK edge.

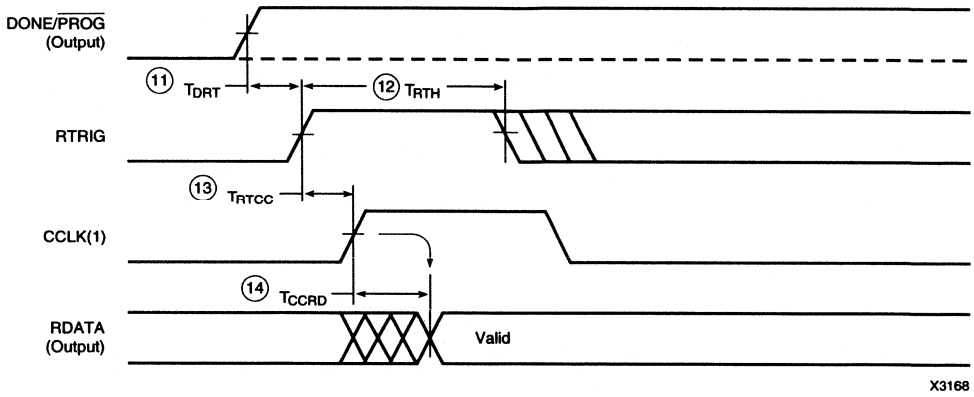
Slave Serial Mode Programming Switching Characteristics



Description		Symbol		Min	Max	Unit
CCLK	To DOUT	1	T_{CCO}		65	ns
	DIN setup	2	T_{DCC}	10		ns
	DIN hold	3	T_{CCD}	40		ns
	High time	4	T_{CCH}	0.25		µs
	Low time	5	T_{CCL}	0.25	5.0	µs
	Frequency		F_{CC}		2	MHz

Note: At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until V_{CC} has reached (2.5 V for the XC2000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require a >1-µs High level on $\overline{\text{RESET}}$, followed by a >6-µs Low level on $\overline{\text{RESET}}$ and D/$\overline{\text{P}}$ after V_{CC} has reached (2.5 V for the XC2000L).

Program Readback Switching Characteristics

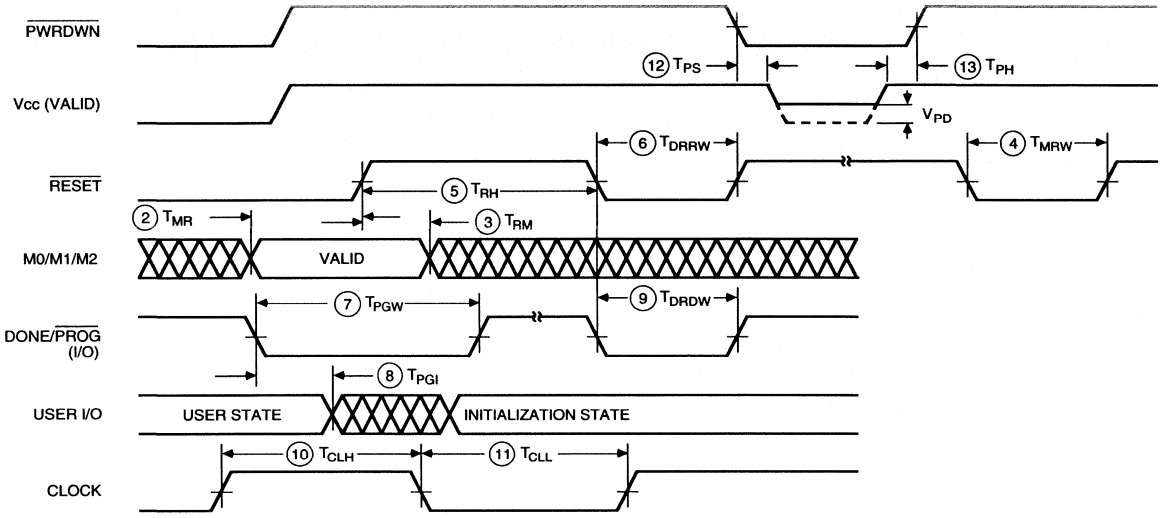


Description		Symbol		Min	Max	Units
RTRIG	PROG setup RTRIG high	11 12	T_{DRT} T_{RTH}	300 250		ns ns
CCLK	RTRIG setup RDATA delay	13 14	T_{RTCC} T_{CCRD}	100	 100	ns ns

Notes: 1. CCLK and DOUT timing are the same as for slave mode, but T_{CCL} for XC2000L is 0.5 µs min.

2. DONE/$\overline{PROG}$ output/input must be HIGH (device programmed) prior to a positive transition of RTRIG (M0).

General LCA Switching Characteristics



	Description	Symbol		Min	Max	Units
RESET[2]	M2, M1, M0 setup	2	T_{MR}	60		ns
	M2, M1, M0 hold	3	T_{RM}	60		ns
	Width—FF Reset	4	T_{MRW}	150		ns
	High before RESET[4]	5	T_{RH}	6		µs
	Device Reset[4]	6	T_{DRRW}	6		µs
DONE/PROG	Progam width (Low)	7	T_{PGW}	6		µs
	Initialization	8	T_{PGI}		7	µs
	Device Reset [4]	9	T_{DRDW}	6		µs
CLOCK	Clock (High)	10	T_{CLH}	8		ns
	Clock (Low)	11	T_{CLL}	8		ns

Notes: 1. At power-up, V_{CC} must rise from 2.0 V to V_{CC} min in less than 25 ms. If this is not possible, configuration can be delayed by holding RESET Low until V_{CC} has reached (2.5 V for the XC2000L). A very long V_{CC} rise time of >100 ms, or a non-monotonically rising V_{CC} may require a >1-µs High level on RESET, followed by a >6-µs Low level on RESET and D/P after V_{CC} has reached (2.5 V for the XC2000L).

2. RESET timing relative to power-on and valid mode lines (M0, M1, M2) is relevant only when RESET is used to delay configuration.
3. Minimum CLOCK widths for the auxillary buffer are 1.25 times the T_{CLH}, T_{CLL}.
4. After RESET is High, RESET = D/P = Low for 6 µs will abort to CLEAR.

Performance

The high performance of the Logic Cell Array results from its patented architectural features and from the use of an advanced high-speed CMOS manufacturing process. Performance may be measured in terms of minimum propagation times for logic elements.

Flip-flop loop delays for the I/O block and logic block flip-flops are about 3 ns. This short delay provides very good performance under asynchronous clock and data conditions. Short loop delays minimize the probability of a metastable condition which can result from assertion of the clock during data transitions. Because of the short loop delay characteristic in the LCA device, the I/O block flip-flops can be used very effectively to synchronize external signals applied to the device. Once synchronized in the I/O block, the signals can be used internally without further consideration of their clock relative timing, except as it applies to the internal logic and routing path delays.

Device Performance

The single parameter which most accurately describes the overall performance of the Logic Cell Array is the maximum toggle rate for a logic block storage element configured as a toggle flip-flop. The configuration for determining the toggle performance of the Logic Cell Array is shown in Figure 17. The clock for the storage element is provided by the global clock buffer and the flip-flop output Q is fed back through the combinatorial logic to form the data input for the next clock edge. Using this arrangement, flip-flops in the Logic Cell Array can be toggled at clock rates from 33–100 MHz, depending on the speed grade used.

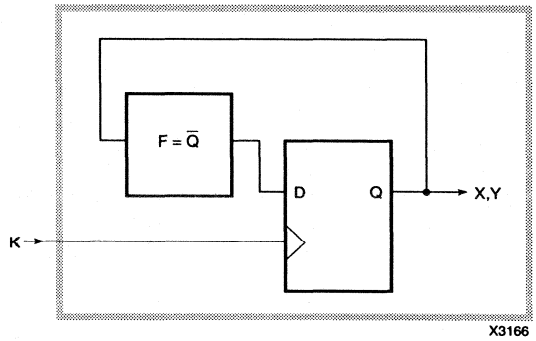


Figure 17. Logic Block Configuration for Toggle Rate Measurement

Actual Logic Cell Array performance is determined by the critical path speed, including both the speed of the logic and storage elements in that path, and the speed of the particular network routing. Figure 18 shows a typical system logic configuration of two flip-flops with an extra combinatorial level between them. To allow the user to make the best use of the capabilities of the device, the delay calculator in the XACT Development System determines worst-case path delays using actual impedance and loading information.

Logic Block Performance

Logic block propagation times are measured from the interconnect point at the input of the combinatorial logic to the output of the block in the interconnect area. Combinatorial performance is independent of logic function

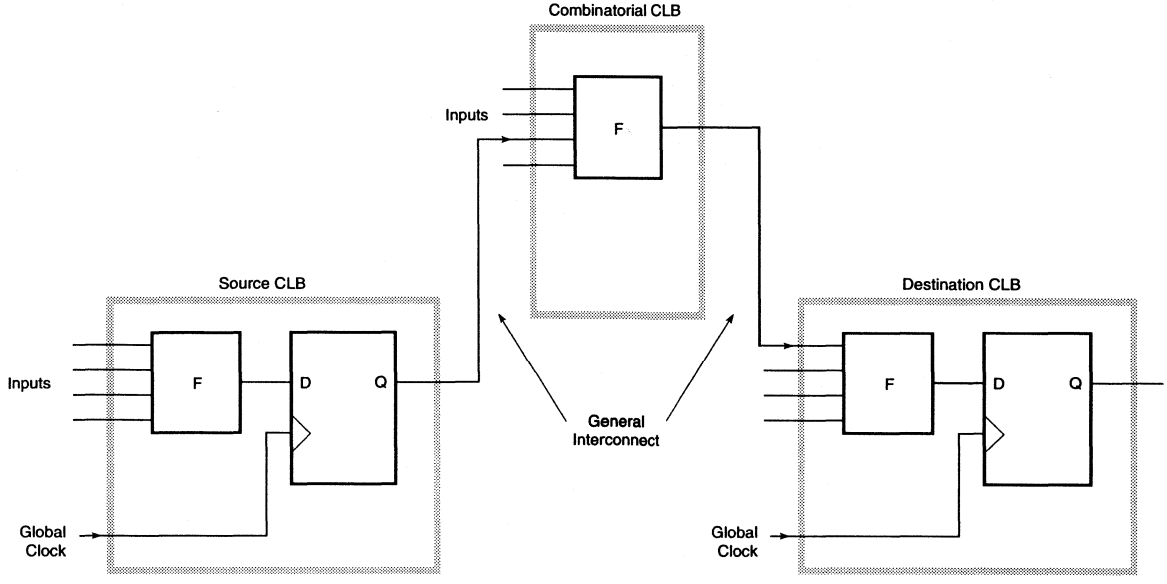


Figure 18. Typical Logic Path

because of the table look-up based implementation. Timing is different when the combinatorial logic is used in conjunction with the storage element. For the combinatorial logic function driving the data input of the storage element, the critical timing is data set-up relative to the clock edge provided to the storage element. The delay from the clock source to the output of the logic block is critical in the timing of signals produced by storage elements. The loading on a logic block output is limited only by the additional propagation delay of the interconnect network. Performance of the logic block is a function of supply voltage and temperature, as shown in Figure 19 .

Interconnect Performance

Interconnect performance depends on the routing resource used to implement the signal path. As discussed earlier, direct interconnect from block to block provides a minimum delay path for a signal.

The single metal segment used for Longlines exhibits low resistance from end to end, but relatively high capacitance. Signals driven through a programmable switch will have the additional impedance of the switch added to their normal drive impedance.

General-purpose interconnect performance depends on the number of switches and segments used, the presence of the bidirectional repowering buffers and the overall loading on the signal path at all points along the path. In calculating the worst-case delay for a general interconnect path, the delay calculator portion of the XACT development system accounts for all of these elements. As an approximation, interconnect delay is proportional to the summation of totals of local metal segments beyond each programmable switch. In effect, the delay is a sum of R-C delays each approximated by an R times the total C it drives. The R of the switch and the C of the interconnect are functions of the particular device performance grade. For a string of three local interconnects, the approximate delay at the first segment, after the first switch resistance, would be three units; an additional two delay units after the next switch plus an additional delay after the last switch in the chain. The interconnect R-C chain terminates at each repowering buffer. Nearly all of the capacitance is in the interconnect metal and switches; the capacitance of the block inputs is not significant.

Power

Power Distribution

Power for the LCA is distributed through a grid to achieve high noise immunity and isolation between logic and I/O. For packages having more than 48 pins, two V_{CC} pins and two ground pins are provided (see Figure 20). Inside the LCA device, a dedicated V_{CC} and ground ring surrounding the logic array provides power to the I/O drivers. An

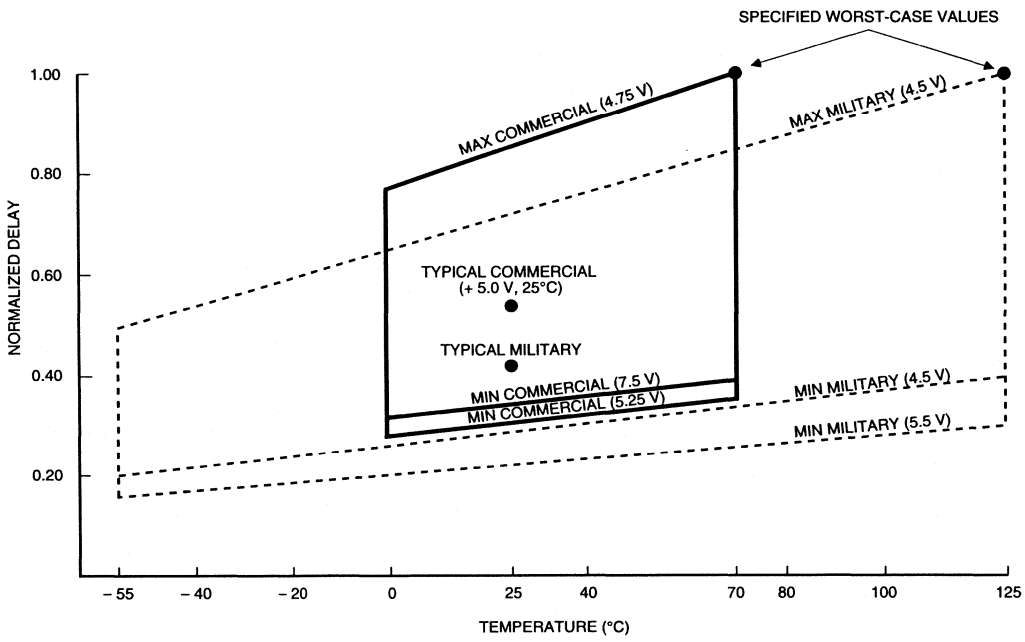
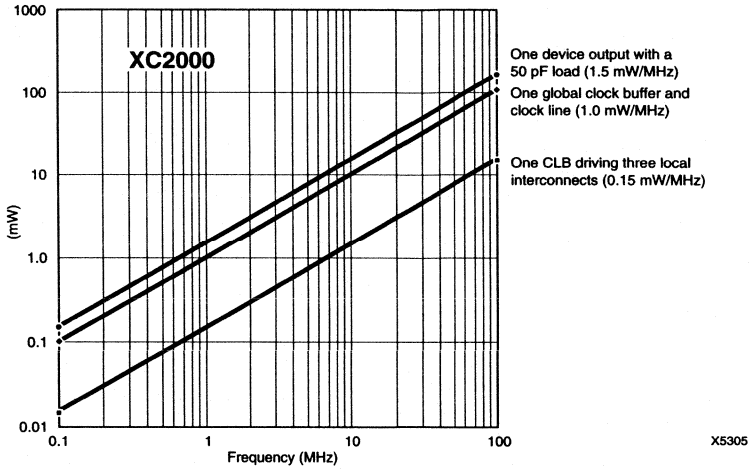


Figure 19. Relative Delay as a Function of Temperature, Supply Voltage and Processing Variations



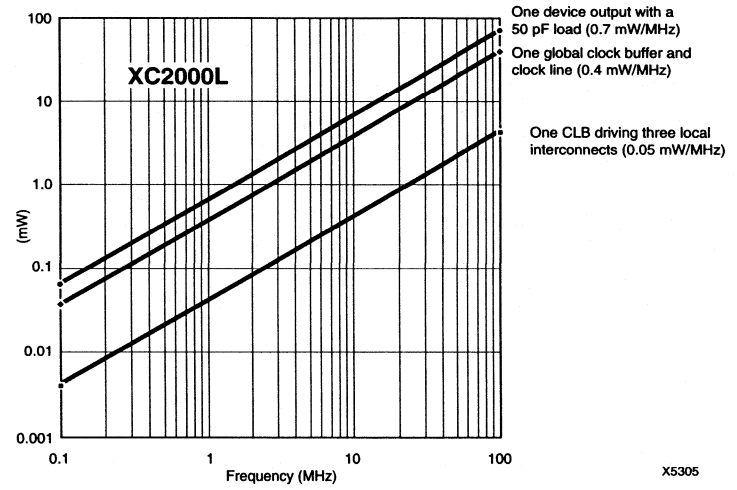


Figure 20. Typical Power Consumption by Element

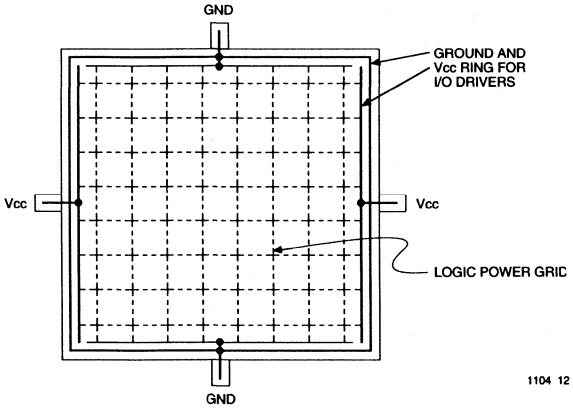


Figure 21. LCA Power Distribution

independent matrix of V_{CC} and ground lines supplies the interior logic of the device. This power distribution grid provides a stable supply and ground for all internal logic, providing the external package power pins are appropriately decoupled. Typically a 0.1 µF capacitor connected between the V_{CC} and ground pins near the package will provide adequate decoupling.

Output buffers capable of driving the specified 4 mA loads under worst-case conditions may be capable of driving 25 to 30 times that current in a best case. Noise can be reduced by minimizing external load capacitance and reducing simultaneous output transitions in the same direction. It may also be beneficial to locate heavily loaded output buffers near the ground pads. Multiple V_{CC} and ground pin connections are required for package types which provide them.

Power Consumption

The Logic Cell Array exhibits the low power consumption characteristic of CMOS ICs. Only quiescent power is required for the LCA configured for CMOS input levels. The TTL input level configuration option requires additional power for level shifting. The power required by the static memory cells which hold the configuration data is very low and may be maintained in a power-down mode.

Typically most of power dissipation is produced by capacitive loads on the output buffers, where the incremental power consumption is 25 µW / pF / MHz . Another component of I/O power is the dc loading on each output pin. For any given system, the user can calculate the I/O power requirement based on the sum of capacitive and resistive loading of the devices driven by the Logic Cell Array.

Internal power supply dissipation is a function of clock frequency and the number of nodes changing on each clock. In an LCA the fraction of nodes changing on a given clock is typically low (10–20%). For example, in a 16-bit binary counter, the average clock produces a change in slightly less than 2 of the 16 bits. In a 4-input AND gate there will be 2 transitions in 16 states. Typical global clock buffer power is about 2.5 mW / MHz for the XC2064 and 3.2 mW / MHz for the XC2018. With a typical load of three general interconnect segments, each Configurable Logic Block output requires about 0.22 mW / MHz of its output frequency. At 3.3 V, the dynamic power consumption is reduced by the square of the voltage ratio, i.e, about 56%.

Dynamic Power Consumption

XC2018 at 5.0V

One CLB driving three local interconnects	0.22 mW/MHz
One device output with a 50-pF load	2.0 mW/MHz
One global clock buffer and line	3.2 mW/MHz

Pin Descriptions

Permanently Dedicated Pins.

V_{CC}
One or two (depending on package type) connections to the nominal +5 V supply voltage. All must be connected.

GND
One or two (depending on package type) connections to ground. All must be connected.

$\overline{PWRDWN}$
A Low on this CMOS-compatible input stops all internal activity, but retains configuration. All flip-flops and latches are reset, all outputs are 3-stated, and all inputs are interpreted as High, independent of their actual level. While $\overline{PWRDWN}$ is Low, V_{CC} may be reduced to any value >2.3 V. When $\overline{PWDWN}$ returns High, the LCA becomes operational with DONE Low for two cycles of the internal 1-MHz clock. During configuration, $\overline{PWRDWN}$ must be High. If not used, $\overline{PWRDWN}$ must be tied to V_{CC}.

$\overline{RESET}$
This is an active Low input which has three functions.

Prior to the start of configuration, a Low input will delay the start of the configuration process. An internal circuit senses the application of power and begins a minimal time-out cycle. When the time-out and $\overline{RESET}$ are complete, the levels of the M lines are sampled and configuration begins.

If $\overline{RESET}$ is asserted during a configuration, the LCA device is re-initialized and restarts the configuration at the termination of $\overline{RESET}$.

If $\overline{RESET}$ is asserted after configuration is complete, it provides a global asynchronous reset of all IOB and CLB storage elements of the LCA device.

$\overline{RESET}$ can also be used to recover from partial power failure. See section on Re-program under "Special Configuration Functions."

CCLK
During configuration, Configuration Clock is an output of an LCA in Master mode or Peripheral mode, but an input in Slave mode. During a Readback, CCLK is a clock input for shifting configuration data out of the LCA.

CCLK drives dynamic circuitry inside the LCA. The Low time may, therefore, not exceed a few microseconds. When used as an input, CCLK must be "parked High". An internal pull-up resistor maintains High when the pin is not being driven.

DONE/$\overline{\text{PROG}}$ (D/$\overline{\text{P}}$)
DONE is an open-drain output, configurable with or without an internal pull-up resistor. At the completion of configuration, the LCA circuitry becomes active in a synchronous order; DONE goes active High one cycle after the IOB outputs go active.

Once configuration is done, a High-to-Low transition of this pin will cause an initialization of the LCA and start a reconfiguration.

M0/RTRIG
As Mode 0, this input and M1, M2 are sampled before the start of configuration to establish the configuration mode to be used.

A Low-to-High input transition, after configuration is complete, acts as a Read Trigger and initiates a Readback of configuration and storage-element data clocked by CCLK. By selecting the appropriate Readback option when generating the bitstream, this operation may be limited to a single Readback, or be inhibited altogether.

M1/$\overline{\text{RDATA}}$
As Mode 1, this input and M0, M2 are sampled before the start of configuration to establish the configuration mode to be used. If Readback is never used, M1 can be tied directly to ground or V_{CC}. If Readback is ever used, M1 must use a 5-kΩ resistor to ground or V_{CC}, to accommodate the $\overline{\text{RDATA}}$ output.

As an active Low Read Data, after configuration is complete, this pin is the output of the Readback data.

User I/O Pins that can have special functions.

M2
During configuration, this input has a weak pull-up resistor. Together with M0 and M1, it is sampled before the start of configuration to establish the configuration mode to be used. After configuration, this pin is a user-programmable I/O pin.

HDC
During configuration, this output is held at a High level to indicate that configuration is not yet complete. After configuration, this pin is a user-programmable I/O pin.

$\overline{\text{LDC}}$
During Configuration, this output is held at a Low level to indicate that the configuration is not yet complete. After configuration, this pin is a user-programmable I/O pin. $\overline{\text{LDC}}$ is particularly useful in Master mode as a Low enable for an EPROM, but it must then be programmed as a High after configuration.

XTL1
This user I/O pin can be used to operate as the output of an amplifier driving an external crystal and bias circuitry.

XTL2
This user I/O pin can be used as the input of an amplifier connected to an external crystal and bias circuitry. The I/O Block is left unconfigured. The oscillator configuration is activated by routing a net from the oscillator buffer symbol output and by the MakeBits program.

$\overline{\text{CS0}}$, $\overline{\text{CS1}}$, CS2, $\overline{\text{WRT}}$
These four inputs represent a set of signals, three active Low and one active High, that are used to control configuration-data entry in the Peripheral mode. Simultaneous assertion of all four inputs generates a Write to the internal data buffer. The removal of any assertion clocks in the D0-D7 data. In Master mode,these pins become part of the parallel configuration byte, D4, D3, D2, D1. After configuration, these pins are user-programmable I/O pins.

$\overline{\text{RCLK}}$
During Master mode configuration $\overline{\text{RCLK}}$ represents a "read" of an external dynamic memory device (normally not used). After configuration, this is a user-programmable I/O pin.

D0–D7
This set of eight pins represents the parallel configuration input for the parallel Master mode. After configuration is complete they are user programmed I/O pins.

A0–A15
During Master Parallel mode, these 16 pins present an address output for a configuration EPROM. After configuration, they are user-programmable I/O pins.

DIN
During Slave or Master Serial configuration, this pin is used as a serial-data input. In the Master or Peripheral configuration, this is the Data 0 input. After configuration, this is a user-programmable I/O pin.

DOUT
During configuration this pin is used to output serial-configuration data to the DIN pin of a daisy-chained slave. After configuration, this is a user-programmable I/O pin.

Unrestricted User I/O Pins.

I/O
An I/O pin may be programmed by the user to be an Input or an Output pin following configuration. All unrestricted I/O pins, plus the special pins mentioned on the following page, have a weak pull-up resistor of 40 to 100 kΩ that becomes active as soon as the device powers up, and stays active until the end of configuration.

XC2064 Configuration Pin Assignments

Configuration Mode: <M2:M1:M0>					44 PLCC	48 DIP	68 PLCC	68 PGA	User Operation
MASTER-SET <0:0:0>	SLAVE <1:1:1>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>						
GND					1		1	B6	GND
<<HIGH>>			A13	(O)			2	A6	I/O
			A6	(O)	2	1	3	B5	
			A12	(O)			4	A5	
			A7	(O)	3	2	5	B4	
			A11	(O)	4	3	6	A4	
			A8	(O)	5	4	7	B3	
			A10	(O)	6	5	8	A3	
			A9	(O)	7	6	9	A2	
PWRDWN (I)					8	7	10	B2	PWR DWN
<<HIGH>>					9	8	11	B1	I/O
					10		12	C2	
					11	9	13	C1	
							14	D2	
						10	15	D1	
							16	E2	
						11	17	E1	
VCC					12	12	18	F2	VCC
<<HIGH>>					13	13	19	F1	I/O
					14		20	G2	
					15	14	21	G1	
							22	H2	
						15	23	H1	
						16	24	J2	
M1 (LOW)	M1 (HIGH)	M1 (LOW)	M1 (HIGH)	M1 (LOW)	16	17	25	J1	RDATA (O)
M0 (LOW)	M0 (HIGH)	M0 (HIGH)	M0 (LOW)	M0 (LOW)	17	18	26	K1	RTRIG (I)
M2 (LOW)	M2 (HIGH)				18	19	27	K2	I/O
HDC (HIGH)					19	20	28	L2	
<<HIGH>>							29	K3	
LDC (LOW)					20	21	30	L3	
<<HIGH>>							31	K4	
					21	22	32	L4	
					22		33	K5	
						23	34	L5	
GND					23	24	35	K6	GND
<<HIGH>>							36	L6	I/O
						25	37	K7	
							38	L7	
						26	39	K8	
						27	40	L8	
			D7 (I)		24	28	41	K9	
			D6 (I)		25	29	42	L9	
					26	30	43		XTL2 OR I/O
RESET (I)					27	31	44		RESET
DONE (O)					28	32	45	K11	PROG (I)
<<HIGH>>					29	33	46	J10	XTL1 OR I/O
							47	J11	I/O
			D5 (I)		30	34	48	H10	
							49	H11	
		CS0 (I)	D4 (I)		31	35	50	G10	
		CS1 (I)	D3 (I)		32	36	51	G11	
Vcc					33		52	F10	VCC
<<HIGH>>							53	F11	I/O
		CS2 (I)	D2 (I)		34	37	54	E10	
							55	E11	
		WRT (I)	D1 (I)		35	38	56	D10	
RCLK			RCLK			39	57	D11	
DIN (I)			D0 (I)		36	40	58	C10	
DOUT (O)					37	41	59	C11	
CCLK (O)	CCLK (I)	CCLK (O)			38	42	60	B11	CCLK (I)
<<HIGH>>			A0	(O)	39	43	61	B10	I/O
			A1	(O)	40	44	62	A10	
			A2	(O)	41	45	63	B9	
			A3	(O)	42	46	64	A9	
			A15	(O)			65	D8	
			A4	(O)	43	47	66	A8	
			A14	(O)			67	B7	
			A5	(O)	44	48	68	A7	

<<HIGH>> is high impedance with a 20-50 k internal pull-up during configuration

X3401

Note: A PLCC in a "PGA-Footprint" socket has a **different** signal pinout than a PGA device.

XC2018 Configuration Pin Assignments

CONFIGURATION MODE: <M2:M1:M0>					44	64	68	84	84	100	USER
MASTER-SER <0:0:0>	SLAVE <1:1:1>	PERIPHERAL <1:0:1>	MASTER-HIGH <1:1:0>	MASTER-LOW <1:0:0>	PLCC	VQFP	PLCC	PLCC	PGA	TQFP	OPERATION
GND					1	8	1	1	C6	13	GND
<<HIGH>>			A13 (O)			9	2	2	A6	14	I/O
								3	A5	15	
								4	B5	16	
			A6 (O)		2	10	3	5	C5	17	
			A12 (O)			11	4	6	A4	18	
			A7 (O)		3	12	5	7	B4	19	
			A11 (O)		4	13	6	8	A3	20	
			A8 (O)		5	14	7	9	A2	21	
			A10 (O)		6	15	8	10	B3	22	
			A9 (O)		7	16	9	11	A1	23	
PWRDWN (I)					8	17	10	12	B2	26	PWR DWN
<<HIGH>>					9	18	11	13	C2	27	I/O
						19	12	14	B1	29	
					10	20	13	15	C1	30	
							14	16	D2	32	
					11	21	15	17	D1	33	
								18	E3	34	
						22	16	19	E2	35	
								20	E1	36	
						23	17	21	F2	37	
Vcc					12	24	18	22	F3	38	Vcc
<<HIGH>>						25	19	23	G3	39	I/O
								24	G1	40	
					13	26	20	25	G2	41	
								26	F1	42	
					14	27	21	27	H1	43	
						28	22	28	H2	45	
					15	29	23	29	J1	47	
						30	24	30	K1	48	
M1 (LOW)	M1 (HIGH)	M1 (LOW)	M1 (HIGH)	M1 (LOW)	16	31	25	31	J2	49	RDATA (O)
M0 (LOW)	M0 (HIGH)	M0 (HIGH)	M0 (LOW)	M0 (LOW)	17	32	26	32	L1	51	RTRIG (I)
M2 (LOW)	M2 (HIGH)				18	33	27	33	K2	53	I/O
HDC (HIGH)					19	34	28	34	K3	54	
<<HIGH>>						35	29	35	L2	55	
LDC (LOW)					20	36	30	36	L3	56	
<<HIGH>>						37	31	37	K4	57	
					21	38	32	38	L4	58	
								39	J5	59	
						39	33	40	K5	60	
						40	34	41	L5	61	
					22			42	K6	62	
GND					23	41	35	43	J6	63	GND
<<HIGH>>								44	J7	64	I/O
						42	36	45	L7	65	
							37	46	K7	66	
						43	38	47	L6	67	
								48	L8	68	
						44	39	49	K8	69	
							40	50	L9	70	
			D7 (I)		24	45	41	51	L10	71	
			D6 (I)		25	46	42	52	K9	72	
					26	47	43	53	L11	73	XTL2 OR I/O
RESET (I)					27	48	44	54	K10	75	RESET
DONE (O)					28	49	45	55	J10	77	PROG (I)
<<HIGH>>					29	50	46	56	K11	78	XTL1 OR I/O
						51	47	57	J11	79	I/O
			D5 (I)		30	52	48	58	H10	80	
								59	H11	82	
						53	49	60	F10	84	
								61	G10	85	
		CSO (I)	D4 (I)		31	54	50	62	G11	86	
		CS1 (I)	D3 (I)		32	55	51	63	G9	87	
Vcc					33	56	52	64	F9	88	Vcc
						57	53	65	F11	89	I/O
		CS2 (I)	D2 (I)		34	58	54	66	E11	90	
<<HIGH>>								67	E10	91	
						59	55	68	E9	92	
								69	D11	93	
		WRT (I)	D1 (I)		35	60	56	70	D10	95	
RCLK			RCLK			61	57	71	C11	96	
	DIN (I)		D0 (I)		36	62	58	72	B11	97	
DOUT (O)					37	63	59	73	C10	98	
CCLK (O)	CCLK (I)	CCLK (O)			38	64	60	74	A11	99	CCLK (I)
<<HIGH>>			A0 (O)		39	1	61	75	B10	2	I/O
			A1 (O)		40	2	62	76	B9	3	
			A2 (O)		41	3	63	77	A10	5	
			A3 (O)		42	4	64	78	A9	6	
			A15 (O)				65	79	B8	7	
			A4 (O)		43	5	66	80	A8	8	
			A14 (O)			6	67	81	B6	9	
								82	B7	10	
								83	A7	11	
			A5 (O)		44	7	68	84	C7	12	

<<HIGH>> IS HIGH IMPEDANCE WITH A 20-50 kΩ INTERNAL PULL-UP DURING CONFIGURATION

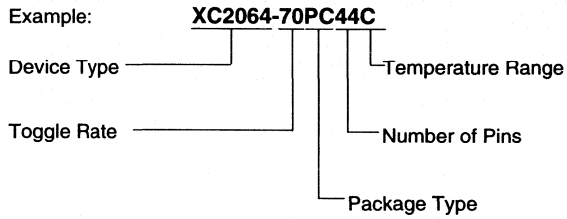
X3461

For a detailed description of the device architecture, see pages 2-187 through 2-195.

For a detailed description of the configuration modes and their timing, see pages 2-200 through 2-208.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Component Availability

PINS		44	48	64	68		84		100	
TYPE		PLAST. PLCC	PLAST. DIP	PLAST. VQFP	PLAST. PLCC	CERAM. PGA	PLAST. PLCC	CERAM. PGA	PLAST. TQFP	PLAST. VQFP
CODE		PC44	PD48	VQ64	PC68	PG68	PC84	PG84	TQ100	VQ100
XC2064	-50		C		C I	C I M				
	-70	C I			C I	C I M				
	-100	C I			C I	C I				
	-130				C					
XC2018	-33							M B		
	-50				C I		C I	C I M B		
	-70	C I			C I		C I	C I M B		
	-100	C I		C I	C I		C I	C I	C I	
	-130			C	C		C	C	C	
XC2064L				(C)	C					
XC2018L				C			C			C

C = Commercial = 0° to +70° C I = Industrial = -40° to +85° C

M = Mil Temp = -55° to +125° C B = MIL-STD-883C Class B

Parentheses indicate future product plans

XC2000
Logic Cell Array Family

Product Specifications

Features

- Fully Field-Programmable:
  - I/O functions
  - Digital logic functions
  - Interconnections
- General-purpose array architecture
- Complete user control of design cycle
- Compatible arrays with logic cell complexity equivalent from 600 to 1,500 gates
- Standard product availability
- 100% factory-tested
- Selectable configuration modes
- Low-power, CMOS, static-memory technology
- Performance equivalent to TTL SSI/MSI
- TTL or CMOS input thresholds
- Complete development system support
  - XACT Design Editor
  - Schematic Entry
  - Macro Library
  - Timing Calculator
  - Logic and Timing Simulator
  - Auto Place/Route

Description

The Logic Cell Array (LCA) is a high density CMOS integrated circuit. Its user-programmable array architecture is made up of three types of configurable elements: Input/Output Blocks, logic blocks and Interconnect. The designer can define individual I/O blocks for interface to external circuitry, define logic blocks to implement logic functions and define interconnection networks to compose larger scale logic functions. The XACT Development System provides interactive graphic design capture and automatic routing. Both logic simulation and in-circuit emulation are available for design verification.

The Logic Cell Array is available in a variety of logic capacities, package styles, temperature ranges and speed grades.

Device	Logic Capacity (gates)	CLBs	User I/O Max	Config. bits
XC2064	600 - 1,000	64	58	12038
XC2018	1,000 - 1,500	100	74	17878

The LCA logic functions and interconnections are determined by data stored in internal static-memory cells. On-chip logic provides for automatic loading of configuration data at power-up. The program data can reside in an EEPROM, EPROM or ROM on the circuit board or on a floppy disk or hard disk. The program can be loaded in a number of modes to accommodate various system requirements.

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

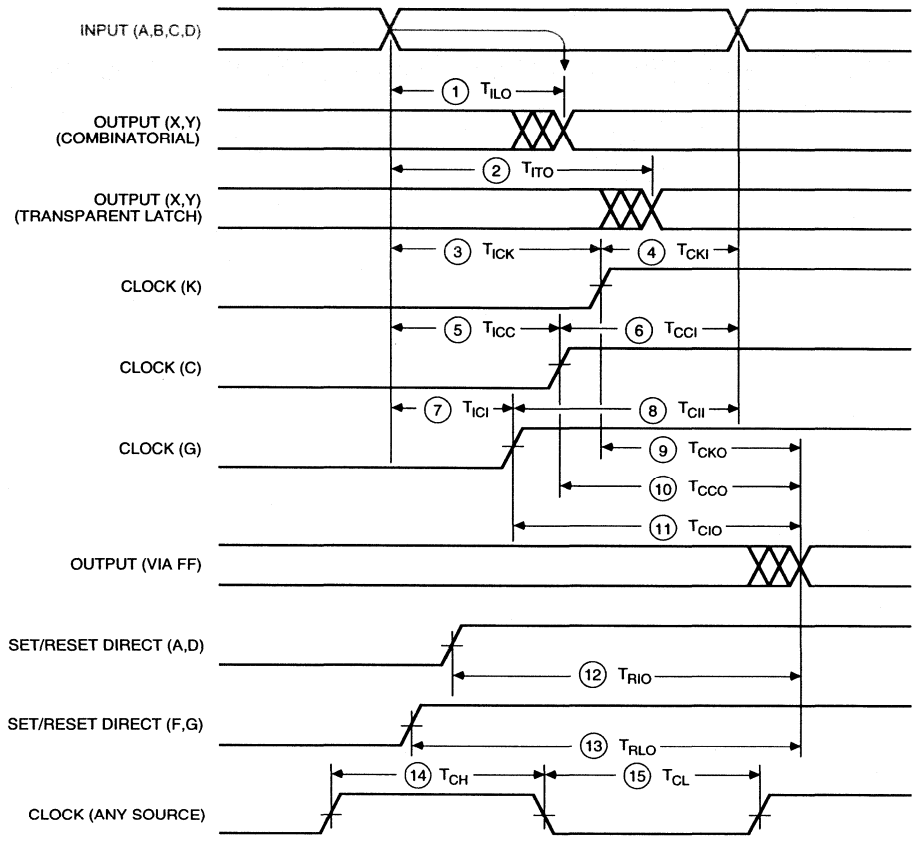
Operating Conditions

Symbol	Description			Min	Max	Units
V_{CC}	Supply voltage relative to GND	Commercial	0°C to +70°C	4.75	5.25	V
	Supply voltage relative to GND	Industrial	–40°C to +85°C	4.5	5.5	V
	Supply voltage relative to GND	Military	–55°C to +125°C	4.5	5.5	V
V_{IHT}	High-level input voltage — TTL configuration			2.0	V_{CC}	V
V_{ILT}	Low-level input voltage — TTL configuration			0	0.8	V
V_{IHC}	High-level input voltage — CMOS configuration			70%	100%	V_{CC}
V_{ILC}	Low-level input voltage — CMOS configuration			0	20%	V_{CC}
T_{IN}	Input signal transition time				250	ns

DC Characteristics Over Operating Conditions

Symbol	Description		Min	Max	Units
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 ma V_{CC} min)	Commercial	3.86		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 ma V_{CC} max)			0.32	V
V_{OH}	High-level output voltage (@ I_{OH} = –4.0 ma V_{CC} min)	Industrial Military	3.76		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 ma V_{CC} max)			0.37	V
V_{CCPD}	Power-down supply voltage ($\overline{PWRDWN}$ must be Low)		2.3		V
I_{CCO}	Quiescent operating power supply current				
	CMOS thresholds (@ V_{CC} Max)			5	mA
	TTL thresholds (@ V_{CC} Max)			12	mA
I_{CCPD}	Power-down supply current ($V_{CC(MAX)}$ @ T_{MAX})			500	µA
I_{IL}	Input Leakage Current		–10	+10	µA
C_{IN}	Input capacitance (sample tested) All Pins except XTL1 and XTL2 XTL1 and XTL2			10 15	pF pF

CLB Switching Characteristic Guidelines



1104 30

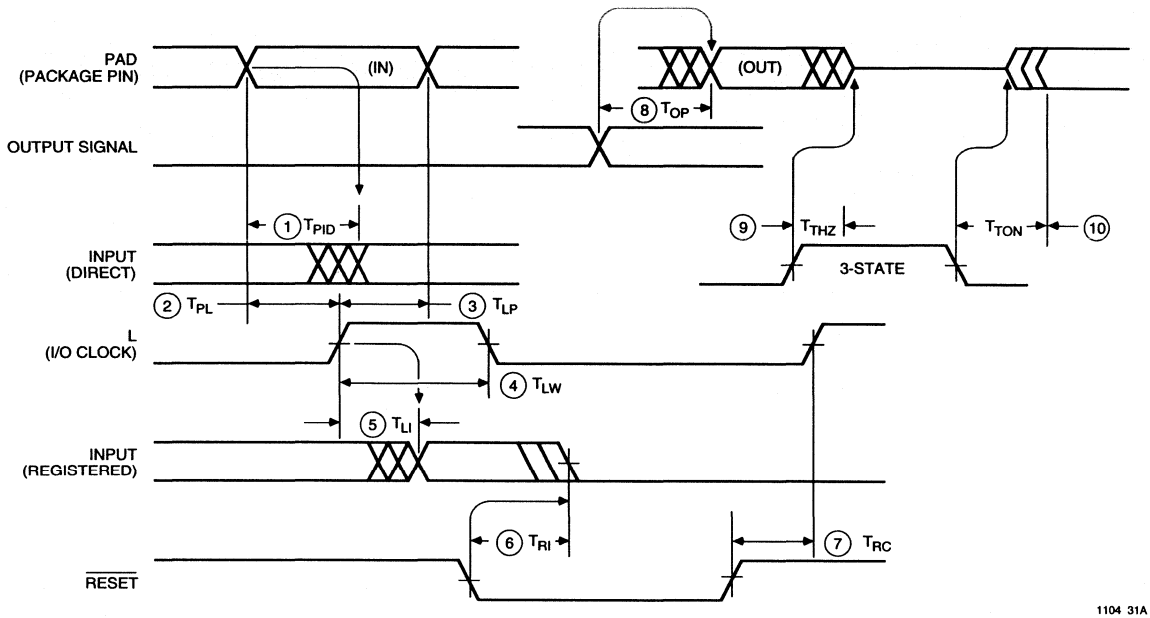
CLB Switching Characteristic Guidelines (Continued)

Speed Grade				-70		-100		-130		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
Logic Input to Output	Combinatorial	1	T_{ILO}		10		8		5.5	ns
	Transparent latch	2	T_{ITO}		14		10		8	ns
	Additional for Q through F or G to out		T_{QLO}		6		6		3.7	ns
K Clock	To output	9	T_{CKO}		10		7		7	ns
	Logic-input setup	3	T_{ICK}	7		6		2.5		ns
	Logic-input hold	4	T_{CKI}	0		0		1.0		ns
C Clock	To output	10	T_{CCO}		13	9			7	ns
	Logic-input setup	5	T_{ICC}	6		5		3		ns
	Logic-input hold	6	T_{CCI}	0		0		1		ns
Logic Input to G Clock	To output	11	T_{CIO}		20		13		13	ns
	Logic-input setup	7	T_{ICI}	3		2		0		ns
	Logic-input hold	8	T_{CII}	4		3		5		ns
Set/Reset direct	Input A or D to output x, y	12	T_{RIO}		16		10		8	ns
	Through F or G to output	13	T_{RLO}		21		14		11	ns
	Reset pad to output x, y		T_{MRQ}		20		17		13	ns
	Separation of set/reset		T_{RS}	7		6		5		ns
	Set/Reset pulse-width		T_{RPW}	7		6		5		ns
Flip-flop Toggle rate	Q through F to flip-flop		F_{CLK}	70		*100		130		MHz
Clock	Clock High	14	T_{CH}	7		*5		3.5		ns
	Clock Low	15	T_{CL}	7		*5		3.5		ns

Notes: 1. All switching characteristics apply to all valid combinations of process, temperature and supply with a nominal chip power dissipation of 250 mW.

* These parameters are for clock pulses generated within a CLB. For an externally generated pulse, derate these parameters by 20%.

IOB Switching Guidelines



	Speed Grade			-70		-100		-130		Units
	Description	Symbol		Min	Max	Min	Max	Min	Max	
Pad (package pin)	To input (direct)	1	T_{PID}		6		4		3.5	ns
I/O Clock	To input (storage)	5	T_{LI}		11		8		7.5	ns
	To pad-input setup	2	T_{PL}	6		4		2		ns
	To pad-input hold	3	T_{LP}	0		0		4		ns
	Pulse width	4	T_{LW}	7		*5		3.5		ns
	Frequency			70		*100				MHz
Output	To pad (output enabled)	8	T_{OP}		9		7		5.5	ns
Three-state	To pad begin hi-Z	9	T_{THZ}		15		11		7	ns
	To pad end hi-Z	10	T_{TON}		15		13		11	ns
$\overline{\text{RESET}}$	To input (storage)	6	T_{RI}		25		17		15	ns
	To input clock	7	T_{RC}	20		14		10		ns

Note: Timing is measured at 0.5 Vcc levels with 50 pF output load.

*These parameters are for clock pulses generated within an LCA device. For an externally applied clock, derate these parameters by 20%.

XC2000L Low-Voltage Logic Cell Array Family

Product Specifications

Features

- Part of the ZERO+ Family of 3.3 V FPGAs
- Low-power, low-supply-voltage FPGA family with two device types
  - JEDEC-compliant 3.3 V version of the XC2000 LCA Family
  - Logic densities from 600 to 1,500 gates
  - Up to 74 user-definable I/Os
- Advanced, low power 0.8 µ CMOS static-memory technology
  - Very low quiescent current consumption, ≤20 µA, 25 times less than XC2000
  - Operating power consumption 66% less than previous generation 5 V FPGAs; 56% less than XC2000
- Identical to the basic XC2000 in structure, pin out, design methodology, and software tools
  - 100% compatible with XC2000 bitstreams
- XC2000L-specific features
  - Guaranteed over the 3.0 to 3.6 V V_{CC} range
  - 4 mA output sink and source current
  - Advanced packaging using thin and very thin quad flat packs

Description

The XC2000L family of FPGAs is optimized for operation from a 3.3 V (nominal) supply. Aside from the electrical and timing parameters listed in this data sheet, the XC2000L family is in all respects identical with the XC2000 family.

The operating power consumption of Xilinx FPGAs is almost exclusively dynamic; it changes with the square of the supply voltage. For a given complexity and clock speed, the XC2000L consumes, therefore, only 44% of the power used by the equivalent XC2000 device. Consistent with its use in battery-powered equipment, the XC2000L family was designed for the lowest possible power-down and quiescent current consumption.

Device	Logic Capacity (gates)	CLBs	User I/O Max	Config. bits
XC2064L	600 - 1,000	64	58	12,038
XC2018L	1,000 - 1,500	100	74	17,878

LCA logic functions and interconnections are determined by data stored in internal static-memory cells. On-chip logic provides for automatic loading of configuration data at power-up. Program data can reside in an EEPROM, EPROM or ROM on the circuit board or on a floppy disk or hard disk. The program can be loaded in a number of modes to accommodate various system requirements.

Xilinx maintains test specifications for each product as controlled documents. To insure the use of the most recently released device performance parameters, please request a copy of the current test-specification revision.

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	–0.5 to +7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	–0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	–65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C
T_J	Junction temperature plastic	+125	°C
	Junction temperature ceramic	+150	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description	Min	Max	Units
V_{CC}	Supply voltage relative to GND (Commercial 0°C to +70°C)	3.0	3.6	V
V_{IH}	High-level input voltage	2.0	V_{CC}+0.3	V
V_{IL}	Low-level input voltage	0.3	0.8	V
T_{IN}	Input signal transition time		250	ns

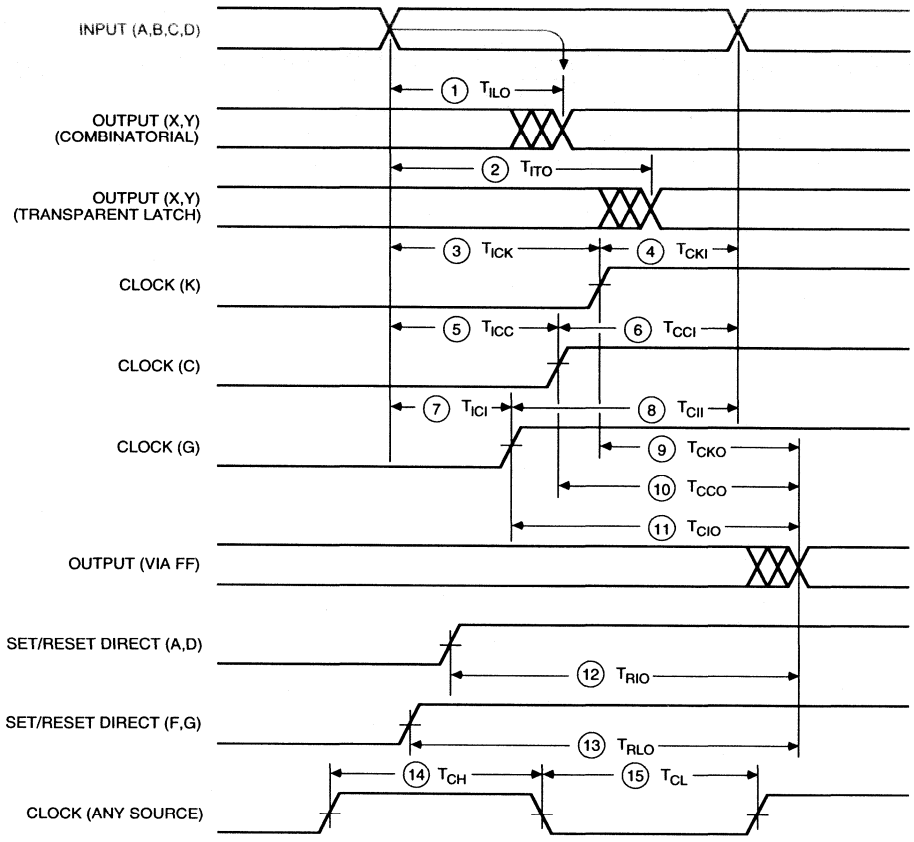
Although the present (1994) devices operate over the full supply voltage range from 3.0 to 5.25 V, Xilinx reserves the right to restrict operation to the 3.0 to 3.6 V range later, when smaller device geometries might preclude operation at 5 V.

DC Characteristics Over Operating Conditions

Symbol	Description	Min	Max	Units
V_{OH}	High-level output voltage (@ I_{OH} = –2.0 mA V_{CC} min)	2.4		V
V_{OL}	Low-level output voltage (@ I_{OL} = 4.0 mA V_{CC} max)		0.4	V
V_{OH}	High-level output voltage (@ –100 µA V_{CC} min)	V_{CC} -0.2		V
V_{OL}	Low-level output voltage (@ 100 µA V_{CC} min)		0.2	V
V_{CCPD}	Power-down supply voltage ($\overline{PWRDWN}$ must be Low)	2.3		V
I_{CCO}	Quiescent operating power supply current*		20	µA
I_{CCPD}	Power-down supply current ($V_{CC(MAX)}$ @ T_{MAX})		10	µA
I_{IL}	Input Leakage Current, all I/O pins in parallel	–10	+10	µA
C_{IN}	Input capacitance (sample tested) All Pins except XTL1 and XTL2		10	pF
	XTL1 and XTL2		15	pF

* With no output current loads, no active input or Longline pull-up resistors, all package pins at V_{CC} or GND, and the LCA device configured with a MakeBits tie option. I_{CCO} is in addition to I_{CCPD}.

CLB Switching Characteristic Guidelines



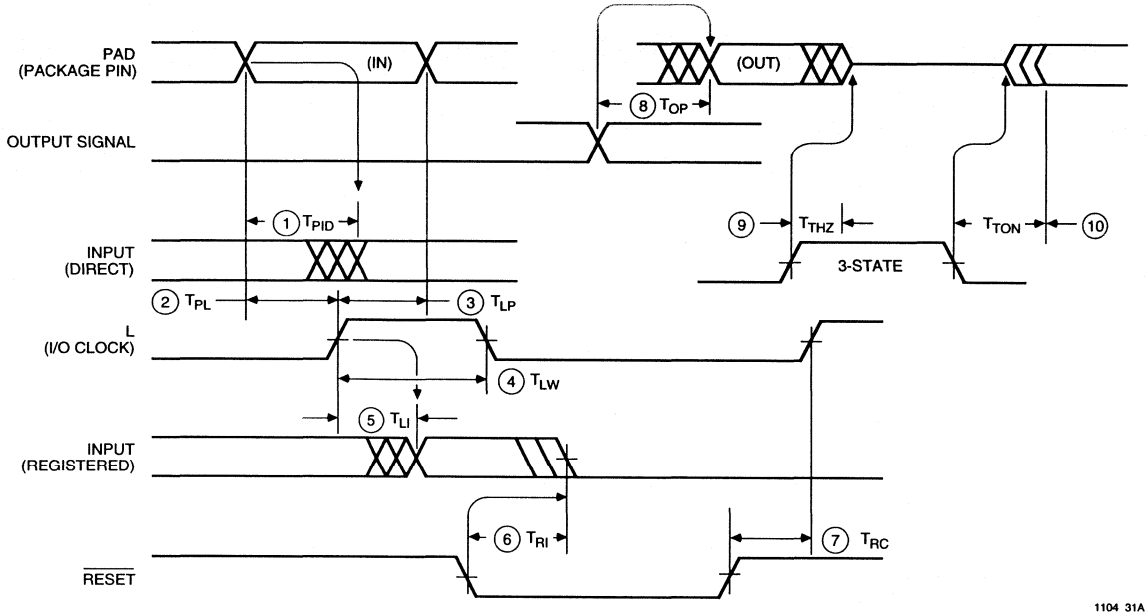
1104 30

CLB Switching Characteristic Guidelines (Continued)

Speed Grade				-10						
	Description	Symbol		Min	Max					Units
Logic Input to Output	Combinatorial Transparent latch Additional for Q through F or G to out	1 2	T_{ILO} T_{ITO} T_{QLO}		9.5 14.0 7.0					ns ns ns
K Clock	To output Logic-input setup Logic-input hold	9 3 4	T_{CKO} T_{ICK} T_{CKI}	 7.0 0	9.5					ns ns ns
C Clock	To output Logic-input setup Logic-input hold	10 5 6	T_{CCO} T_{ICC} T_{CCI}	 3.5 0	13.0					ns ns ns
Logic Input to G Clock	To output Logic-input setup Logic-input hold	11 7 8	T_{CIO} T_{ICI} T_{CII}	 0 5.0	20.0					ns ns ns
Set/Reset direct	Input A or D to output x, y Through F or G to output Reset pad to output x, y Separation of set/reset Set/Reset pulse-width	12 13	T_{RIO} T_{RLO} T_{MRQ} T_{RS} T_{RPW}	 7.0 7.0	10.0 17.0 20.0					ns ns ns ns ns
Flip-flop Toggle rate	Q through F to flip-flop		F_{CLK}	70.0						MHz
Clock	Clock High Clock Low	14 15	T_{CH} T_{CL}	7.0 7.0						ns ns

Notes: 1. All switching characteristics apply to all valid combinations of process, temperature and supply with a nominal chip power dissipation of 250 mW.

IOB Switching Guidelines



Speed Grade				-10						
	Description	Symbol		Min	Max					Units
Pad (package pin)	To input (direct)	1	T_{PID}		7.0					ns
I/O Clock	To input (storage) To pad-input setup To pad-input hold Pulse width Frequency	5 2 3 4	T_{LI} T_{PL} T_{LP} T_{LW}	 5.0 1.0 7.0	10.0					ns ns ns ns MHz
Output	To pad (output enabled)	8	T_{OP}		10.5					ns
Three-state	To pad begin hi-Z To pad end hi-Z	9 10	T_{THZ} T_{TON}		15.0 20.0					ns ns
$\overline{\text{RESET}}$	To input (storage) To input clock	6 7	T_{RI} T_{RC}	 20.0	25.0					ns ns

Note: Timing is measured at 0.5 V_{CC} levels with 50 pF output load.

XC17000 Family of Serial Configuration PROMs

Table of Contents

Overview

Serial Configuration PROMs

Xilinx offers several pin- and function-compatible serial one-time-programmable PROMs in plastic and ceramic packages.

The original family consists of the XC1736A, XC1765 followed by the XC17128.

The XC1736A is the only serial PROM that lacks the programmable Reset polarity option, and the XC17128 is the only serial PROM that can be clocked at the full 10 MHz required by the XC4000 in fast configuration mode. All other serial PROMs can be clocked at up to 5 MHz.

In early 1993, Xilinx introduced the D-series of serial PROMs, the XC1718D, XC1736D, and XC1765D, all with programmable Reset polarity, improved ESD protection, and all with max 5 MHz clock frequency. These devices are programmed with a lower voltage and a different programming algorithm than the older parts. The user needs the appropriate update from the programmer vendor. These devices will become the mainstream serial PROMs, and, beyond the traditional packages, they are also available in the space-saving SO8 package.

In 1994, Xilinx will also ship the L-series of serial PROMs, the XC1718L and XC1765L. These devices operate at the new industry-standard low supply voltage of 3.3 V (3.0 to 3.6 V).

Capacity

Device	Configuration Bits
XC1718D or L	18,144
XC1736D or L	36,288
XC1765D or L	65,536
XC17128	131,072

polarity control

Component Availability

Pins	8			20
Type	Plast. DIP	Ceram. DIP	Plast. SOIC	Plast. PLCC
Code	PD8	DD8	SO8/VO8	PC20
XC1718D	CI	M	CI	CI
XC1736D	CI	M	CI	CI
XC1765D	CI	MR	CI	CI
XC1718L	(C)		(C)	(C)
XC1765L	(C)		(C)	(C)
XC17128	C*	M		C*

C = Commercial = 0° to +70°C M = Mil Temp = -55° to +125°C
I = Industrial = -40° to +85°C R = High Rel = -55° to +125°C
*XC17128 C = 0° to +70°C or -40° to +85°C

XC17000 Family of Serial Configuration PROMs

Preliminary Product Specifications

Features

- Extended family of one-time programmable (OTP) bit-serial read-only memories used for storing the configuration bitstreams of Xilinx FPGAs
- On-chip address counter, incremented by each rising edge on the clock input
- Simple interface to the FPGA requires only one user I/O pin
- Cascadable for storing longer or multiple bitstreams
- Programmable reset polarity (active High or active Low) for compatibility with different FPGA solutions, (the older XC1736A has active-High reset only)
- XC17128 supports XC4000 fast configuration mode (10 MHz)
- Low-power CMOS EPROM process
- Available in 5 V and 3.3 V versions
- Available in plastic and ceramic packages, and commercial, industrial and military temperature ranges
- Space efficient 8-pin DIP, 8-pin SOIC or 20-pin surface-mount packages.
- Programming support by leading programmer manufacturers.

Description

The XC17000 family of serial configuration PROMs (SCPs) provides an easy-to-use, cost-effective method for storing Xilinx FPGA configuration bitstreams.

When the FPGA is in master serial mode, it generates a configuration clock that drives the SCP. A short access time after the rising clock edge, data appears on the SCP DATA output pin that is connected to the FPGA DIN pin. The FPGA generates the appropriate number of clock pulses to complete the configuration. Once configured, it disables the SCP. When the FPGA is in slave mode, the SCP and the FPGA must both be clocked by an incoming signal.

Multiple devices can be concatenated by using the $\overline{CEO}$ output to drive the $\overline{CE}$ input of the following device. The clock inputs and the DATA outputs of all SCPs in this chain are interconnected. All devices are compatible and can be cascaded with other members of the family.

For device programming, the XACT development system compiles the LCA design file into a standard Hex format, which is then transferred to the programmer.

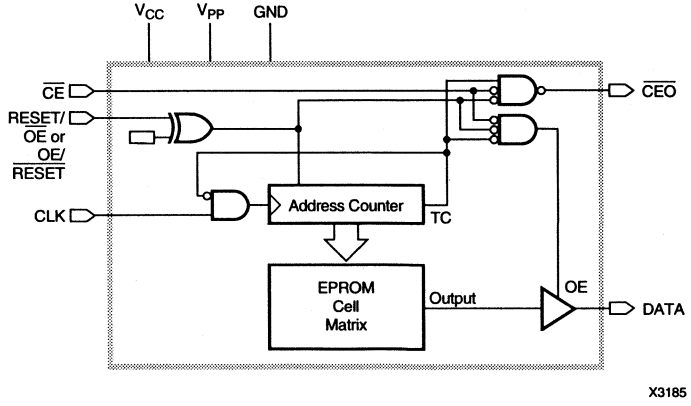


Figure 1. Simplified Block Diagram (does not show programming circuit)

Pin Assignments

DATA
Data output, 3-stated when either $\overline{CE}$ or $\overline{OE}$ are inactive. During programming, the DATA pin is I/O. Note that $\overline{OE}$ can be programmed to be either active High or active Low.

CLK
Each rising edge on the CLK input increments the internal address counter, if both $\overline{CE}$ and $\overline{OE}$ are active. Note that $\overline{OE}$ can be programmed to be either active High or active Low.

RESET/$\overline{OE}$
When High, this input holds the address counter reset and 3-states the DATA output. The polarity of this input pin is programmable as either RESET/$\overline{OE}$ or OE/$\overline{RESET}$. To avoid confusion, this document describes the pin as RESET/$\overline{OE}$, although the opposite polarity is possible on all devices except the older XC1736A.

$\overline{CE}$
When High, this pin disables the internal address counter, 3-states the DATA output, and forces the device into low-I_{CC} standby mode.

$\overline{CEO}$
Chip Enable output, to be connected to the CE input of the next SCP in the daisy chain. This output is Low when the $\overline{CE}$ and $\overline{OE}$ inputs are both active AND the internal address counter has been incremented beyond its Terminal Count (TC) value. In other words: when the PROM has been read, $\overline{CEO}$ will follow $\overline{CE}$ as long as $\overline{OE}$ is active. When $\overline{OE}$ goes inactive, $\overline{CEO}$ stays High until the PROM is reset. Note that $\overline{OE}$ can be programmed to be either active High or active Low.

V_{PP}
Programming voltage. No overshoot above the specified max voltage is permitted on this pin. For normal read operation, this pin *must* be connected to V_{CC}. Failure to do so may lead to unpredictable, temperature-dependent operation and severe problems in circuit debugging. *Do not leave V_{PP} floating!*

V_{CC}
Positive supply pin.

GND
Ground pin

Serial PROM Pinouts

Pin Name	8-Pin	20-Pin
DATA	1	2
CLK	2	4
RESET/$\overline{OE}$ (OE/$\overline{RESET}$)	3	6
$\overline{CE}$	4	8
GND	5	10
$\overline{CEO}$	6	14
V_{PP}	7	17
V_{CC}	8	20

Capacity

Device	Configuration Bits
XC1718D or L	18,144
XC1736D or L	36,288
XC1765D or L	65,536
XC17128	131,072

plus 32 bits for reset polarity control

Number of Configuration Bits, Including Header for all Xilinx FPGAs and Compatible SCP Type

Device	Configuration Bits	SCP
XC2064	12,038	XC1718
XC2018	17,878	XC1718
XC3020/3120	14,819	XC1718
XC3030/3130	22,216	XC1736
XC3042/3142	30,824	XC1736
XC3064/3164	46,104	XC1765
XC3090/3190	64,200	XC1765
XC3195	94,984	XC17128
XC4002A	31,668	XC1736
XC4003A	45,676	XC1765
XC4003H	53,967	XC1765
XC4004A	62,244	XC1765
XC4005A	81,372	XC17128
XC4005/4005H	95,000	XC17128
XC4006	119,832	XC17128
XC4008	147,544	XC17128 + XC1718
XC4010	178,136	XC17128 + XC1765
XC4013	247,960	XC17128 + XC17128
XC4025	422,168	XC17128 + XC17128 + XC17128 + XC1736

Controlling Serial PROMs

Most connections between the LCA device and the Serial PROM are simple and self-explanatory.

- The DATA output of the PROM drives DIN of the LCA devices.
- The master LCA CCLK output drives the CLK input of the Serial PROM.
- The $\overline{CEO}$ output of any Serial PROM can be used to drive the $\overline{CE}$ input of the next serial PROM in a cascade chain of PROMs.
- V_{PP} ***must*** be connected to V_{CC}. Leaving V_{PP} open can lead to unreliable, temperature-dependent operation.

There are, however, two different ways to use the inputs $\overline{CE}$ and $\overline{OE}$.

1. The LCA D/$\overline{P}$ or LDC output drives both $\overline{CE}$ and $\overline{OE}$ in parallel. This is the simplest connection, but it fails if a user applies $\overline{RESET}$ during the LCA configuration process. The LCA device aborts the configuration and then restarts a new configuration, as intended, but the Serial PROM does not reset its address counter, since it never saw a High level on its $\overline{OE}$ input. The new configuration, therefore, reads the remaining data in the PROM and interprets it as preamble, length count etc. Since the LCA device is the master, it issues the necessary number of CCLK pulses, up to 16 million (2^{24}) and D/$\overline{P}$ goes High. However, the LCA configuration will be completely wrong, with potential contentions inside the LCA device and on its output pins. *This method must, therefore, never be used when there is any chance of external reset during configuration.*
2. The LCA D/$\overline{P}$ or LDC output drives only the $\overline{CE}$ input of the Serial PROM while its $\overline{OE}$ input is driven by the LCA $\overline{RESET}$ input. This connection works under all normal circumstances, even when the user aborts a configuration before D/$\overline{P}$ has gone High. The Low level on the OE input during reset clears the PROM internal address pointer, so that the reconfiguration starts at the beginning. The reset polarity should be inverted for this mode to be used. It is strongly recommended that this method, shown in Figure 2, be used whenever possible.

LCA Master Serial Mode Summary

The I/O and logic functions of the Logic Cell Array and their associated interconnections are established by a configuration program. The program is loaded either automatically upon power up, or on command, depending on the state of the three LCA mode pins. In Master Mode, the Logic Cell Array automatically loads the configuration program from an external memory. The Serial Configuration PROM has been designed for compatibility with the Master Serial Mode.

Upon power-up or reconfiguration, an LCA device enters the Master Serial Mode whenever all three of the LCA mode-select pins are Low (M0=0, M1=0, M2=0). Data is read from the Serial Configuration PROM sequentially on a single data line. Synchronization is provided by the rising edge of the temporary signal CCLK, which is generated during configuration.

Master Serial Mode provides a simple configuration interface. Only a serial data line and two control lines are required to configure an LCA device. Data from the Serial Configuration PROM is read sequentially, accessed via the internal address and bit counters which are incremented on every valid rising edge of CCLK.

If the user-programmable, dual-function DIN pin on the LCA device is used only for configuration, it must still be held at a defined level during normal operation. The XC3000 and XC4000 families take care of this automatically with an on-chip default pull-up resistor. With XC2000-family devices, the user must either configure DIN as an active output, or provide a defined level, e.g., by using an external pull-up resistor, if DIN is configured as an input.

Programming the LCA With Counters Unchanged Upon Completion

When multiple LCA-configurations for a single LCA are stored in a Serial Configuration PROM, the $\overline{OE}$ pin should be tied Low as shown in Figure 3. Upon power-up, the internal address counters are reset and configuration begins with the first program stored in memory. Since the $\overline{OE}$ pin is held Low, the address counters are left unchanged after configuration is complete. Therefore, to reprogram the LCA with another program, the D/$\overline{P}$ line is pulled Low and configuration begins at the last value of the address counters.

Cascading Serial Configuration PROMs

For multiple LCAs configured as a daisy-chain, or for future LCAs requiring larger configuration memories, cascaded SCPs provide additional memory. After the last bit from the first SCP is read, the next clock signal to the SCP asserts its $\overline{CEO}$ output Low and disables its DATA line. The second SCP recognizes the Low level on its $\overline{CE}$ input and enables its DATA output. See Figure 2.

After configuration is complete, the address counters of all cascaded SCPs are reset if the LCA $\overline{RESET}$ pin goes Low, assuming the SCP reset polarity option has been inverted.

If the address counters are not to be reset upon completion, then the RESET/$\overline{OE}$ inputs can be tied to ground, as shown in Figure 3. To reprogram the LCA device with another program, the D/$\overline{P}$ line goes Low and configuration begins where the address counters had stopped. In this case, avoid contention between DATA and the configured I/O use of DIN.

When more than a few SCPs are daisy-chained, the designer must evaluate the worst-case CCLK-to-DATA delay resulting from the cascaded $\overline{CE}$-to-$\overline{CEO}$ delays. All Xilinx LCA devices require valid input data a set-up time before the next rising CCLK edge.

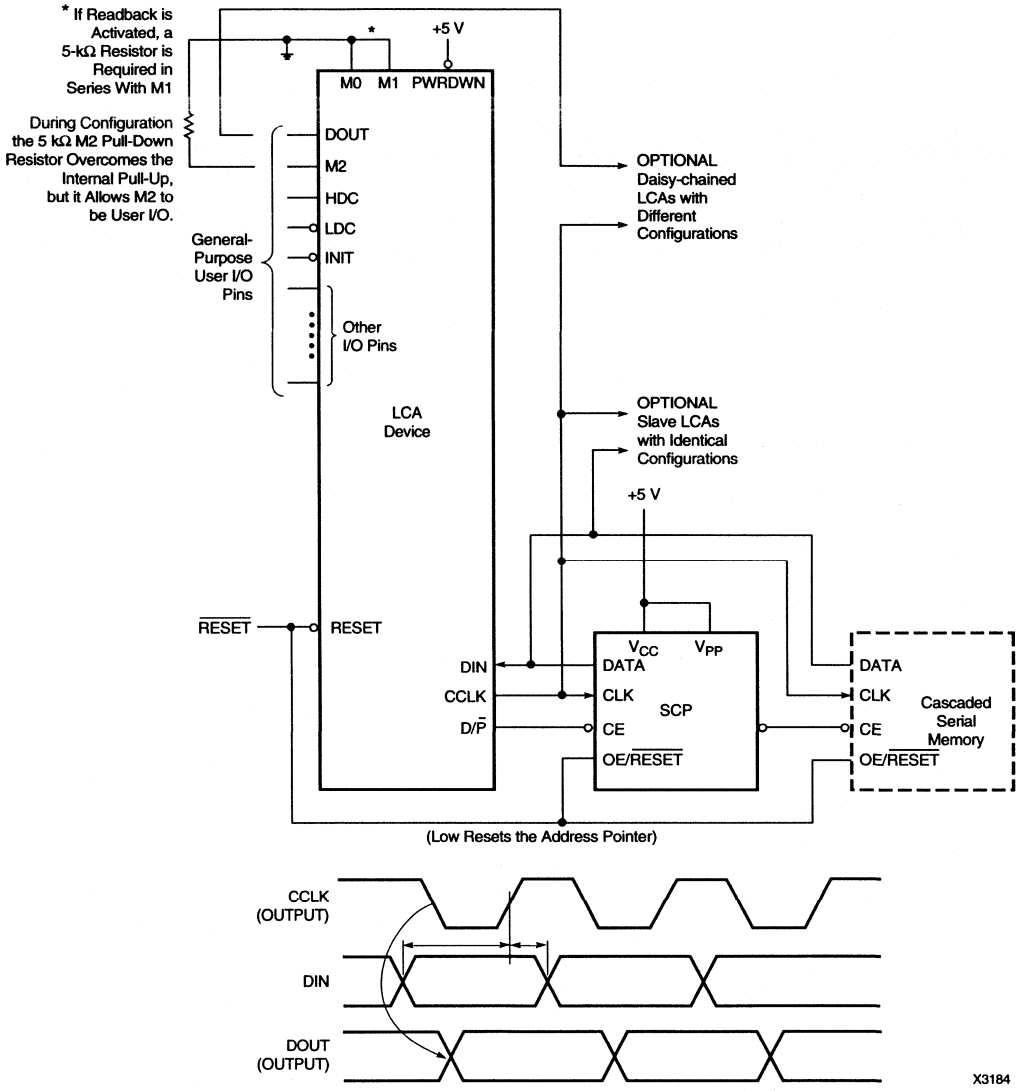
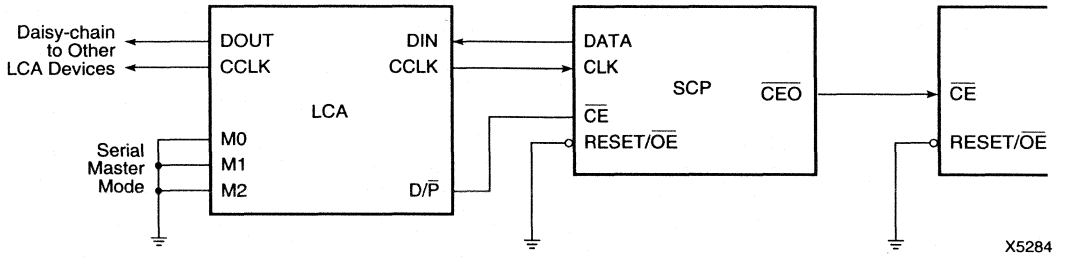


Figure 2. Master Serial Mode. The one-time-programmable Serial Configuration PROM supports automatic loading of configuration programs. Multiple devices can be cascaded to support additional LCA devices. An early D/$\overline{P}$ inhibits the PROM data output one CCLK cycle before the LCA I/Os become active.



Notes: 1. If programmed for active High Reset, tie RESET to V_{CC}.
2. If M2 is tied directly to ground, it should be programmed as an input during operation.
3. If the LCA is reset during configuration, it will abort back to initialization state. An external signal is then required to reset the XC17XX counters.

Figure 3. Address Counters Not Reset at the End of Configuration

Standby Mode

The PROM enters a low-power standby mode whenever $\overline{CE}$ is asserted High. In this mode, the SCP consumes less than 0.5 mA of current. The output remains in a high-impedance state regardless of the state of the $\overline{OE}$ input.

Reducing Standby Current to Zero

The 0.5 mA of serial PROM standby current may be unacceptable in a low-current application. It is, however, possible to achieve zero standby current by disconnecting the PROM ground lead from system ground and connecting it to the $\overline{LDC}$ pin of the LCA, as shown in Figure 4.

As a result, the PROM powers up together with the LCA, since $\overline{LDC}$ goes Low immediately after power-up; the PROM then stays powered-up until the end of the configuration process. When the user outputs go active, $\overline{LDC}$ must go 3-state and thus cut off the PROM supply current. $\overline{LDC}$ must, therefore, be configurated as an input with pull-up resistor, not as an active High output.

The PROM operating current (typically <5 mA) causes a voltage drop of typically 100 mV on the $\overline{LDC}$ output, reducing the PROM supply voltage by that amount. This violates the specification, but is guaranteed to work, since all PROMs are factory-tested at 4.5 V V_{CC}. Multiple PROMs increase the $\overline{LDC}$ sink current by only 0.5 mA per additional PROM.

$\overline{LDC}$ must never be active High, because there might be a few more CCLK pulses at the end of configuration, which will pull the PROM's CLK input below the level of the PROM ground pin. In user mode, it is, therefore, important to avoid driving the PROM with any active High or Low levels. That means that the $\overline{LDC}$ and DIN pins cannot be used in user mode, they must both be configurated as inputs with a pull-up resistor. The $\overline{CE}$ input must be tied to the SCP ground pin. $\overline{RESET}$ (active Low) must be connected to the LCA $\overline{RESET}$ input.

This design assumes that only one configuration bitstream is stored in one or multiple PROMs. It is inherently impossible to use this design when multiple bitstreams are stored in one PROM or one daisy chain of PROMs.

Programming the XC17000 Family Serial PROMs

The devices can be programmed on programmers supplied by Xilinx or qualified third-party vendors. The user must ensure that the appropriate programming algorithm and voltage are used. Different product types use different algorithms and voltages, and the wrong choice can permanently damage the device.

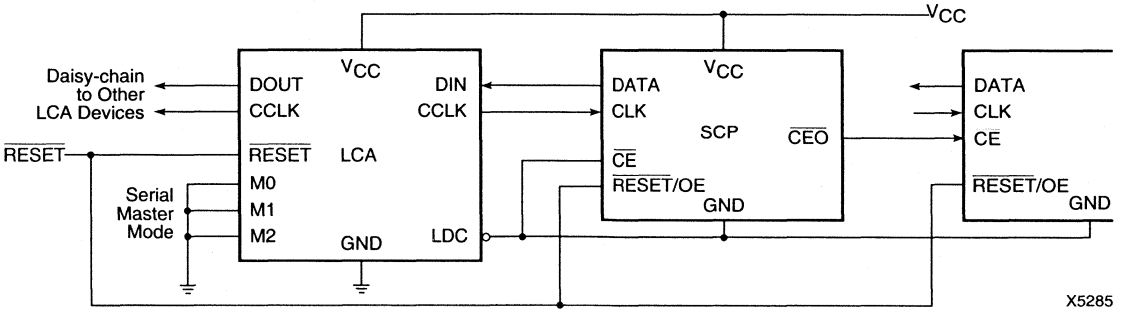


Figure 4. Zero-Standby Current Circuit

XC1718D, XC1736D, XC1765D, XC17128

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	-0.5 to +7.0	V
V_{PP}	Supply voltage relative to GND: XC1718D, XC1736D, XC1765D	-0.5 to +12.5	V
	Supply voltage relative to GND: XC17128	-0.5 to +15.5	V
V_{IN}	Input voltage relative to GND	-0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	-0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	-65 to +125	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Description		Min	Max	Units
V_{CC}	Commercial	Supply voltage relative to GND -0 °C to +70°C	4.75	5.25	V
	Industrial	Supply voltage relative to GND -40°C to +85°C	4.5	5.5	V
	Military	Supply voltage relative to GND -55°C to +125°C	4.5	5.5	V

DC Characteristics Over Operating Condition

Symbol	Description		Min	Max	Units
V_{IH}	High-level input voltage		2.0	V_{CC}	V
V_{IL}	Low-level input voltage		0	0.8	V
V_{OH}	High-level output voltage (I_{OH} = -4 mA)	Commercial	3.86		V
V_{OL}	Low-level output voltage (I_{OL} = +4 mA)			0.32	V
V_{OH}	High-level output voltage (I_{OH} = -4 mA)	Industrial	3.76		V
V_{OL}	Low-level output voltage (I_{OL} = +4 mA)			0.37	V
V_{OH}	High-level output voltage (I_{OH} = -4 mA)	Military	3.7		V
V_{OL}	Low-level output voltage (I_{OL} = +4 mA)			0.4	V
I_{CCA}	Supply current, active mode			10	mA
I_{CCS}	Supply current, standby mode			0.5	mA
I_L	Input or output leakage current		-10	10	µA

Note: During normal read operation V_{PP} ***must*** be connected to V_{CC}

XC1718L and XC1765L

Absolute Maximum Ratings

Symbol	Description		Units
V_{CC}	Supply voltage relative to GND	-0.5 to +6.0	V
V_{PP}	Supply voltage relative to GND	-0.5 to +12.5	V
V_{IN}	Input voltage with respect to GND	-0.5 to V_{CC} +0.5	V
V_{TS}	Voltage applied to 3-state output	-0.5 to V_{CC} +0.5	V
T_{STG}	Storage temperature (ambient)	-65 to +125	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in.)	+260	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

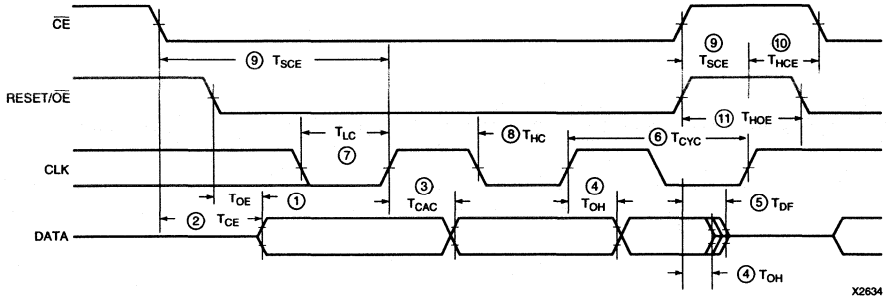
Symbol	Description		Min	Max	Units
V_{CC}	Commercial	Supply voltage relative to GND -0 °C to +70°C	3.0	3.6	V
	Industrial	Supply voltage relative to GND -40°C to +85°C	3.0	3.6	V
	Military	Supply voltage relative to GND -55°C to +125°C	3.0	3.6	V

DC Characteristics Over Operating Condition

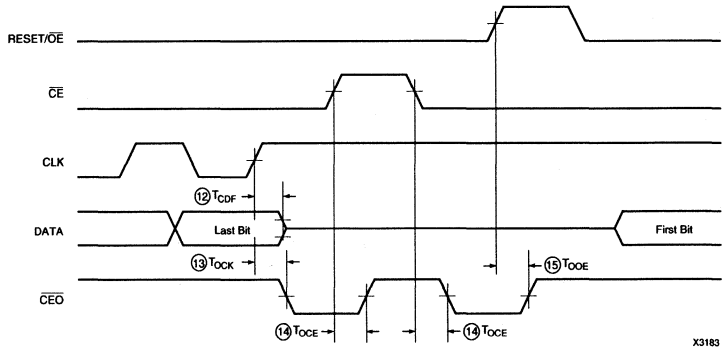
Symbol	Description	Min	Max	Units
V_{IH}	High-level input voltage	2.0	V_{CC}	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{OH}	High-level output voltage (I_{OH} = -4 mA)	2.4		V
V_{OL}	Low-level output voltage (I_{OL} = +4 mA)		0.4	V
I_{CCA}	Supply current, active mode		5	mA
I_{CCS}	Supply current, standby mode		0.5	mA
I_L	Input or output leakage current	-10	10	µA

Note: During normal read operation V_{PP} ***must*** be connected to V_{CC}

AC Characteristics Over Operating Conditions



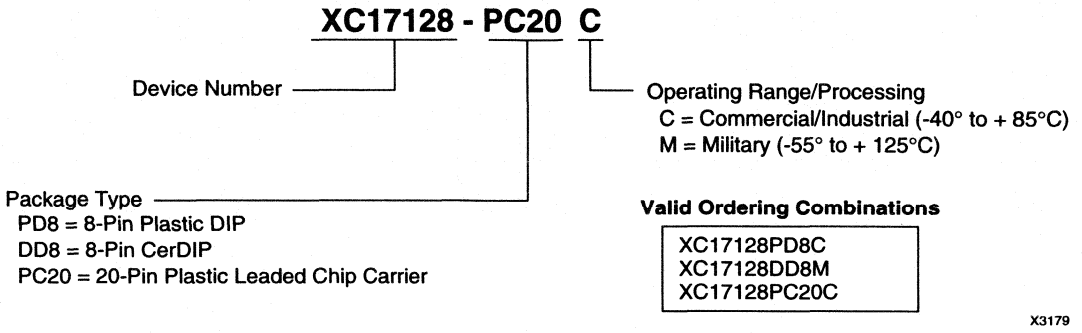
Symbol		Description	XC1718D, XC1736D, XC1765D		XC1718L, XC1765L		XC17128		Units
			Min	Max	Min	Max	Min	Max	
1	T_{HOE}	$\overline{OE}$ to Data Delay		45		45		50	ns
2	T_{CE}	$\overline{CE}$ to Data Delay		60		60		50	ns
3	T_{CAC}	CLK to Data Delay		150		200		60	ns
4	T_{OH}	Data Hold From $\overline{CE}$, $\overline{OE}$, or CLK	0		0		0		ns
5	T_{DF}	$\overline{CE}$ or $\overline{OE}$ to Data Float Delay[2]		50		50		50	ns
6	T_{CYC}	Clock Periods	200		400		100		ns
7	T_{LC}	CLK Low Time[3]	100		100		25		ns
8	THC	CLK High Time[3]	100		100		25		ns
9	TSCE	$\overline{CE}$ Setup Time to CLK (to guarantee proper counting)	25		40		25		ns
10	T_{HCE}	$\overline{CE}$ Hold Time to CLK (to guarantee proper counting)	0		0		0		ns
11	T_{HOE}	$\overline{OE}$ High Time (guarantees counters are reset)	100		100		20		n

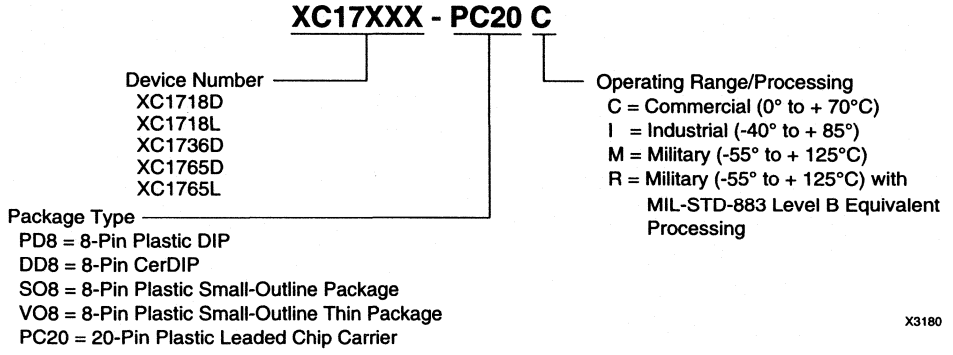


Symbol		Description	XC1718D, XC1736D, XC1765D		XC1718L, XC1765L		XC17128		Units
			Min	Max	Min	Max	Min	Max	
12	T_{CDF}	CLK to Data Float Delay[2]		50		50		50	ns
13	T_{OCK}	CLK to $\overline{CEO}$ Delay		65		65		40	ns
14	T_{OCE}	CE to $\overline{CEO}$ Delay		45		45		40	ns
15	T_{OOE}	RESET/$\overline{OE}$ to $\overline{CEO}$ Delay		40		40		45	ns

Notes: 1. AC test load = 50 pF
2. Float delays are measured with minimum tester ac load and maximum dc load.
3. Guaranteed by design, not tested.
4. All ac parameters are measured with V_{IL} = 0.0 V and V_{IH} = 3.0 V.

Ordering Information





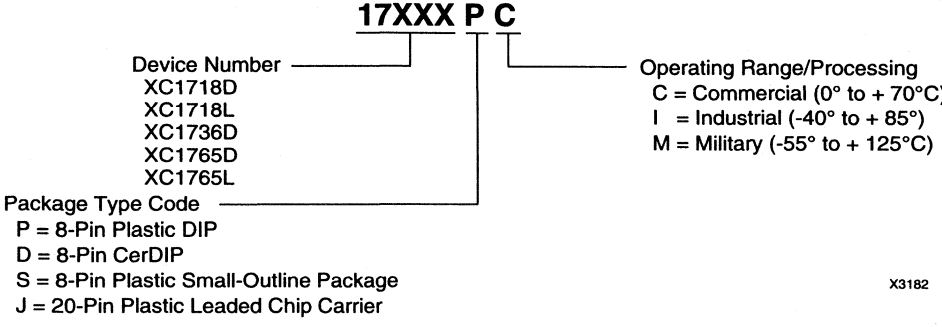
Valid Ordering Combinations

XC1718DPD8C	XC1736DPD8C	XC1765DPD8C	XC1718LPD8C	XC1765LPD8C
XC1718DPD8I	XC1736DPD8I	XC1765DPD8I	XC1718LSO8C	XC1765LSO8C
XC1718DSO8C	XC1736DSO8C	XC1765DSO8C	XC1718LVO8C	XC1765LVO8C
XC1718DVO8C	XC1736DVO8C	XC1765DVO8C	XC1718LPC20C	XC1765LPC20C
XC1718DSO8I	XC1736DSO8I	XC1765DSO8I		
XC1718DVO8I	XC1736DVO8I	XC1765DVO8I		
XC1718DPC20C	XC1736DPC20C	XC1765DPC20C		
XC1718DPC20I	XC1736DPC20I	XC1765DPC20I		
	XC1736DDD8M	XC1765DDD8M		
		XC1765DDD8R		

X3181

Marking Information

Due to the small size of the serial PROM package, the complete ordering part number cannot be marked on the package. The XC prefix is deleted and the package code is simplified. Device marking is as follows.



Overview

In the XC7000 family, Xilinx offers two evolutionary and compatible generations of Erasable Programmable Logic Devices (EPLDs). Xilinx EPLDs combine the advantages of LSI-high-level of integrations, small size, low cost, high-reliability – with the user's need to create applications-specific circuits, without incurring the cost, delay, and risk of mask-programmable gate arrays.

Every Xilinx EPLD provides multiple programmable logic structures, called Function Blocks (FBs), interconnected together through an unrestricted Universal Interconnect Matrix (UIM). Each FB contains nine Macrocells driven by a programmable AND/OR array. Any device input and any Macrocell output can be connected to the input of any other Macrocell. This unrestricted programmable interconnect structure, guarantees 100% routability. In addition, the familiar AND/OR logic of the traditional PAL architecture, makes Xilinx EPLDs easy to use.

The delay through a device is predictable. Any function that can be implemented in one Function Block will run at the specified device speed.

Xilinx offers the industry's only *low-cost*, fully-functional EPLD design software. XEPLD Translator software allows the user to create, implement, and verify digital logic circuits targeting the full range of XC7000 devices.

The EPLD devices are based on a state-of-the-art CMOS EPROM technology and are 100% tested over all operating conditions.

XC7300 EPLD Family

Table of Contents

Overview

Introduced in 1993, the XC7300 EPLD family is designed to address customer needs for high performance and high density in a single complex programmable logic device. The XC7300 features an innovative Dual Block architecture consisting of two types of Functions Blocks (FBs) interconnected by a Universal Interconnect Matrix (UIM). The Fast Function Blocks are optimized for high performance and High Density Function Blocks for highest possible logic density. This innovative Dual-Block architecture combined with the 100% interconnect capability of the UIM, makes the XC7300 family ideal for new CPLD designs and PAL conversion.

Xilinx XC7300 family offers four advantages.

- Dual-Block architecture optimized for speed and density
- Unrestricted Universal Interconnect Matrix (UIM) for guaranteed interconnect
- Dedicated high-speed arithmetic carry logic for efficient implementation of fast adders, subtractors, accumulators, and magnitude comparator
- Mixed voltage I/O operation providing 3.3 V or 5 V interface configurations

Component Availability

Pins		44		68		84			144	160	184	225	
Type		Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Ceramic PGA	Ceramic PGA	Plastic PQFP	Ceramic PGA	Plastic BGA	Ceramic BGA
Code		PC44	WC44	PC68	WC68	PC84	WC84	PG84	PG144	PQ160	PG184	PB225	WB225
XC7336	-15	CI	CI(M)										
	-12	CI	CI(M)										
	-10	CI	CI										
	-7	C	C										
XC7354	-15	CI	CI	CI	CI(M)								
	-12	CI	CI	CI	CI(M)								
	-10	CI	CI	CI	CI								
XC7372	-15			CI	CI	CI	CI(M)	(CI)					
	-12			CI	CI	CI	CI(M)	(CI)					
	-10			CI	CI	CI	CI	(CI)					
XC73108	-20					CI	CI(M)		CI(M)	CI		(CI)	(CI)
	-15					CI	CI(M)		CI(M)	CI		(CI)	(CI)
	-12					CI	CI		CI	CI		(CI)	(CI)
	-10					(C)	(C)		(C)	(C)		(C)	(C)
XC73144	-15										(CI)	(CI)	(CI)
	-12										(CI)	(CI)	(CI)
	-10										(C)	(C)	(C)

X3479

Product Description

Features

- High-performance Erasable Programmable Logic Devices (EPLDs)
  - 7.5 to 12 ns pin-to-pin delays
  - 80 to 125 MHz maximum clock frequency
- Advanced Dual-Block architecture
  - Fast Function Blocks
  - High-Density Function Blocks
- 100% interconnect matrix
- High-speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 51 MHz 18-bit accumulators
- Multiple independent clocks
- Each input programmable as direct, latched, or registered
- High-drive 24 mA output
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- Power management options
- Multiple security bits for design protection
- Supported by industry standard design and verification tools
- Advanced 0.8µ CMOS EPROM process

Description

The XC7300 family employs a unique Dual-Block architecture, which provides high speed operations via Fast Function Blocks and/or high density capability via High Density Function Blocks.

Fast Function Blocks (FFBs) provide fast, pin-to-pin speed and logic throughput for critical decoding and ultra-fast state machine applications. High-density Function Blocks (FBs) provide maximum logic density and system-level features to implement complex functions with predictable timing for adders and accumulators, wide functions and state machines requiring large numbers of product terms, and other forms of complex logic.

In addition, the XC7300 architecture employs the Universal Interconnect Matrix (UIM) which guarantees 100% interconnect of all internal functions. This interconnect scheme provides constant, short interconnect delays for all routing paths through the UIM. Constant interconnect delays simplify device timing and guarantee design performance, regardless of logic placement within the chip.

All XC7300 devices are designed in 0.8µ CMOS EPROM technology.

All XC7300 EPLDs include programmable power management features to specify high-performance or low-power operation on an individual Macrocell-by-Macrocell basis. Unused Macrocells are automatically turned off to mini-

The XC7300 Family

	XC7336	XC7354	XC7372	XC73108	XC73144
Typical 22V10 Equivalent	4	6	8	12	16
Number of Macrocells	36	54	72	108	144
Number of Function Blocks	4	6	8	12	16
Number of Flip-Flops	36	108	126	198	234
Number of Fast Inputs	18	24	30	42	54
Number of Signal Pins	38	66	84	120	156

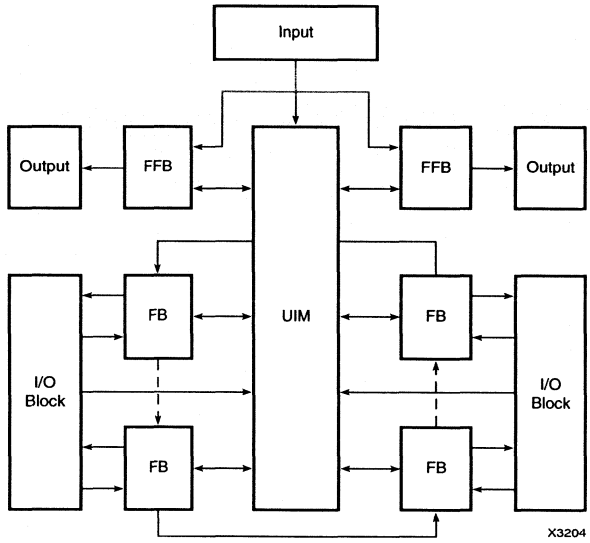


Figure 1. XC7300 Device Block Diagram

mize power dissipation. Designers can operate speed-critical paths at maximum performance, while non-critical paths dissipate less power.

Xilinx development software supports XC7300 EPLD design using third-party schematic entry tools, HDL compilers, or direct equation-based text files. Using a PC or a workstation and one of these design capture methods, designs are automatically mapped to an XC7300 EPLD in a matter of minutes.

The XC7300 devices are available in plastic and ceramic leaded chip carriers, pin-grid-array (PGA), and quad flat pack (QFP) packages. Package options include both windowed ceramic for design prototypes and one-time programmable plastic versions for cost-effective production volume.

Architecture

The XC7300 architecture consists of multiple programmable Function Blocks interconnected by a UIM as shown in Figure 1. The Dual-Block architecture contains two types of function blocks: Fast Function Blocks and High-Density Function Blocks. Both types of function blocks, and the I/O blocks, are interconnected through the UIM.

Fast Function Blocks

The Fast Function Block has 24 inputs which can be individually selected from the UIM, 12 fast input pins, or the nine Macrocell feedbacks from the Fast Function Block. The programmable AND array in each Fast Function Block generates 45 product terms to drive the nine Macrocells in each Fast Function Block. Each Macrocell can be configured for registered or combinatorial logic. See Figure 2.

Five product terms from the programmable AND array are allocated to each Macrocell. Four of these product terms are ORed together and may be optionally inverted before driving the input of a programmable D-type flip-flop. The fifth product term drives the asynchronous active-High programmable Reset or Set Input to the Macrocell flip-flop. The flip-flop can be configured as a D-type or Toggle flip-flop, or transparent for combinatorial outputs.

Two fast function block Macrocell differences exist when comparing the XC7336 FFB to the XC7354, XC7372 and XC73108 FFBs.

In the XC7336, five product terms from the programmable AND array are allocated to each Macrocell. Four of these product-terms are OR'd together and may be optionally inverted before driving the input of a programmable D-type flip-flop. The fifth product-term drives the asynchronous active High programmable Set or Reset input to the Macrocell flip-flop. The flip-flop can be configured as a D-type or Toggle flip-flop, or transparent for combinatorial outputs. See Figure 2.

In the XC7354, XC7372 and XC73108, five product terms from the programmable AND array are allocated to each Macrocell. Four of these product-terms are OR'd together, inverted and drive the input of a programmable D-type flip-flop. The fifth product-term drives the asynchronous active High programmable Set input to the Macrocell flip-flop. The flip-flop can be configured as a D-type flip-flop or transparent for combinatorial outputs. See Figure 3.

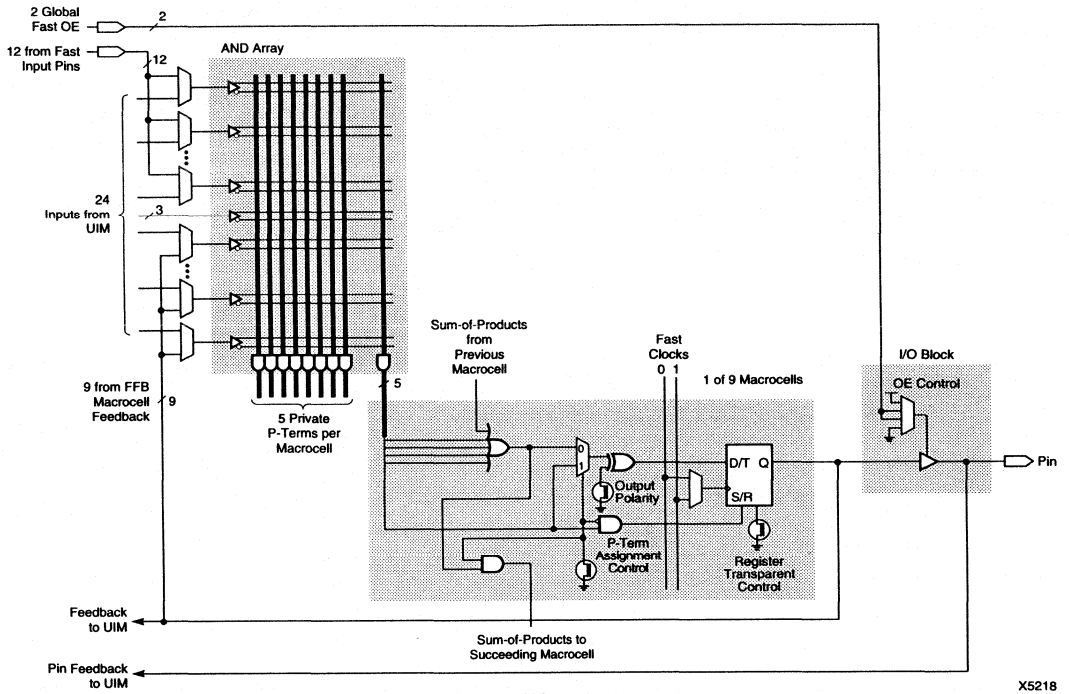


Figure 2. Fast Function Block and Macrocell Schematic for the XC7336

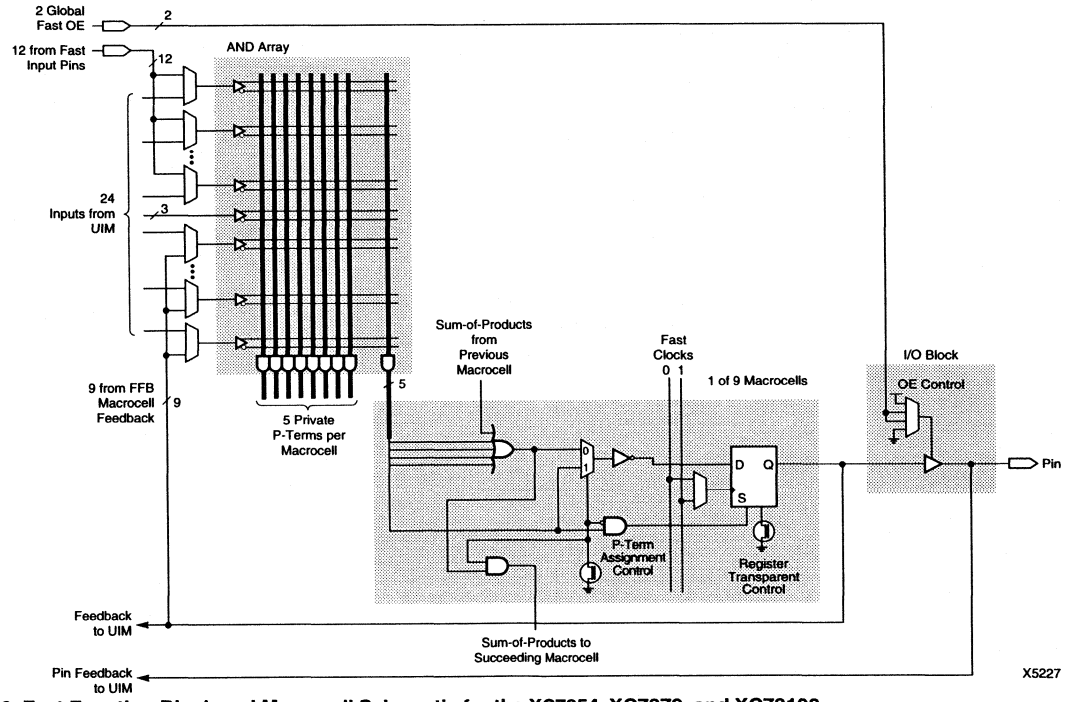


Figure 3. Fast Function Block and Macrocell Schematic for the XC7354, XC7372, and XC73108

The programmable clock source is one of two global Fast-Clock signals (FCLK0 or FCLK1) that are distributed with short delay and minimal skew over the entire chip.

The Fast Function Block Macrocells drive chip outputs directly through 3-state output buffers. Each output buffer can be individually controlled by one of two dedicated Fast Output Enable inputs or permanently enabled or disabled. The Macrocell output can also be routed back as an input to the Fast Function Block and the UIM.

Product Term Assignment

Each Macrocell sum-of-product OR gates can be expanded using the Fast Function Block product term assignment scheme. Product-term assignment transfers product-terms in increments of four product-terms from one Macrocell to the neighboring Macrocell (Figure 4). Complex logic functions requiring up to 36 product-terms can be implemented using all nine Macrocells within the Fast Function Block. When product-terms are assigned to adjacent Macrocells, the product-term normally dedicated to the Set or Reset function becomes the input to the Macrocell register.

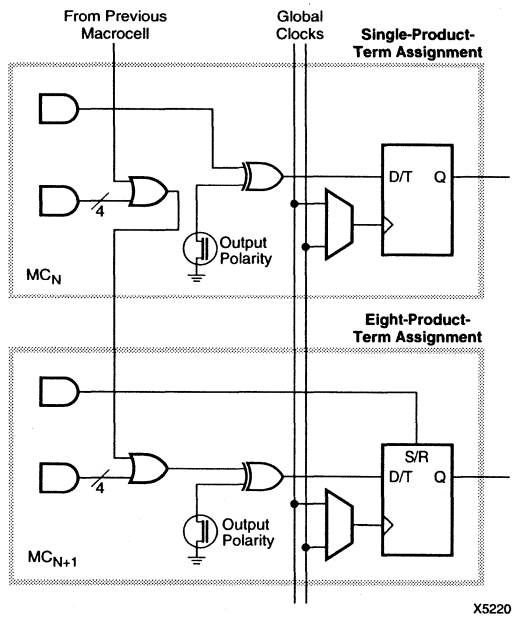


Figure 4. Fast Function Block Product-Term Assignment

High-Density Function Blocks

Each member of the XC7300 family contains multiple, High-Density Function Blocks linked though the UIM. Each Function Block contains nine Macrocells. Each Macrocell can be configured for either registered or combinatorial logic. A detailed block diagram of the XC7300 FB is shown in Figure 5.

Each FB receives 21 signals and their complements from the UIM and an additional three inputs from the Fast Input (FI) pins.

Shared and Private Product Terms

Each Macrocell contains five private product terms that can be used as the primary inputs for combinatorial functions implemented in the Arithmetic Logic Unit (ALU), or as individual Reset, Set, Output-Enable, and Clock logic functions for the flip-flop. Each Function Block also provides an additional 12 shared product terms, which are uncommitted product terms available for any of the nine Macrocells within the Function Block.

Four private product terms can be ORed together with up to four shared product terms to drive the D1 input to the ALU. The D2 input is driven by the OR of the fifth private product term and up to eight of the remaining shared product terms. The shared product terms add no logic delay, and each shared product term can be connected to one or all nine Macrocells in the Function Block.

Arithmetic Logic Unit

The functional versatility of each Macrocell is enhanced through additional gating and control functions available in the ALU. A detailed block diagram of the XC7300 ALU is shown in Figure 6.

The ALU has two programmable modes; *logic* and *arithmetic*. In logic mode, the ALU functions as a 2-input function generator using a 4-bit look-up table that can be programmed to generate any Boolean function of its D1 and D2 inputs as illustrated in Table 1.

The function generator can OR its inputs, widening the OR function to a maximum of 17 inputs. It can AND them, which means that one sum-of-products can be used to mask the other. It can also XOR them, toggling the flip-flop or comparing the two sums of products. Either or both of the sum-of-product inputs to the ALU can be inverted, and either or both can be ignored.

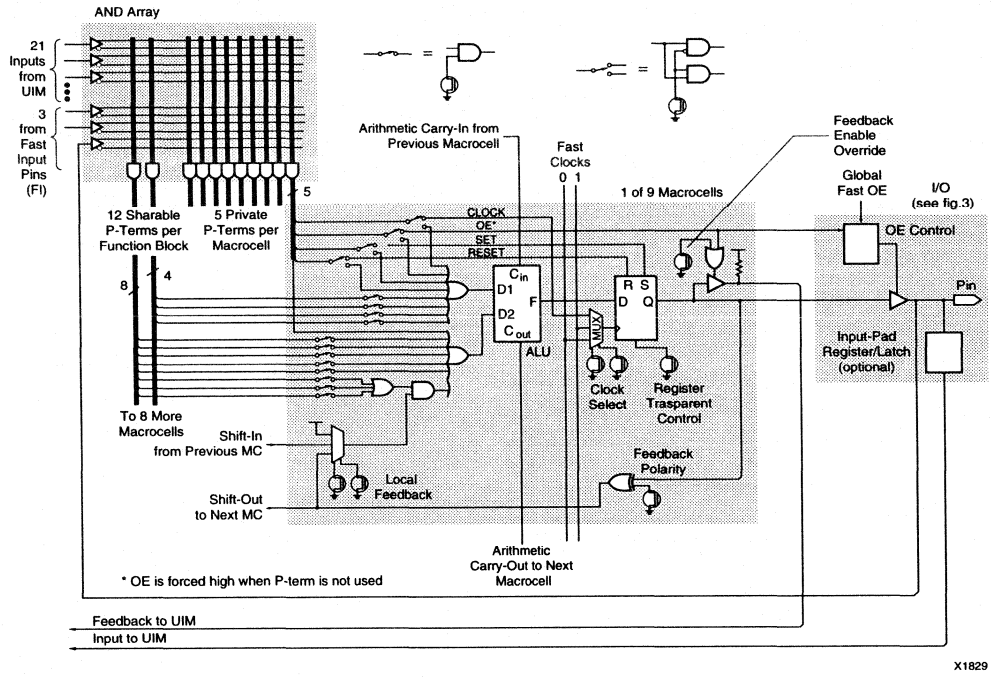


Figure 5. High-Density Function Block and Macrocell Schematic

Table 1. Function Generator Logic Operations

Function	
D1:+: D2	$\overline{D1}$:+: D2
D1 * D2	$\overline{D1}$ * $\overline{D2}$
D1 + D2	$\overline{D1}$ + $\overline{D2}$
D1	D2
$\overline{D1}$	$\overline{D2}$
D1 * $\overline{D2}$	$\overline{D1}$ * D2
D1 + $\overline{D2}$	$\overline{D1}$ + D2

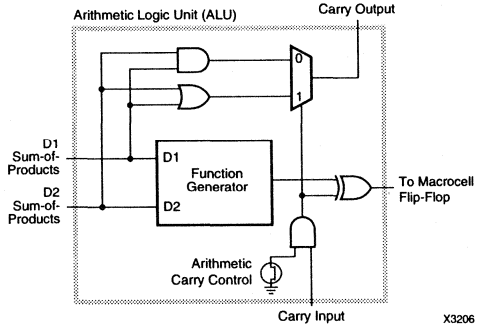


Figure 6. ALU Schematic

Therefore, the ALU can implement one additional layer of logic without any speed penalty.

In arithmetic mode, the ALU block can be programmed to generate the arithmetic sum or difference of the D1 and D2 inputs. Combined with the carry input from the next lower Macrocell, the ALU operates as a 1-bit full adder generating a carry output to the next higher Macrocell. The carry chain propagates between adjacent Macrocells and also crosses the boundaries between Function Blocks. This dedicated carry chain overcomes the inherent speed and density problems of the traditional EPLD architecture when trying to perform arithmetic functions.

Carry Lookahead

Each Function Block provides a carry lookahead generator capable of anticipating the carry across all nine Macrocells. The carry lookahead generator reduces the ripple-carry delay of wide arithmetic functions such as add, subtract, and magnitude compare to that of the first nine bits, plus the carry lookahead delay of the higher-order Function Blocks.

Macrocell Flip-Flop

The ALU block output drives the input of a programmable D-type flip-flop. The flip-flop is triggered by the rising edge of the clock input, but it can be configured as transparent,

making the Q output identical to the D input, independent of the clock, or as a conventional flip-flop.

The Macrocell clock source is programmable and can be one of the private product terms or one of two global FastCLK signals (FCLK0 and FCLK1). Global FastCLK signals are distributed to every Macrocell flip-flop with short delay and minimal skew.

The asynchronous Set and Reset product terms override the clocked operation. If both asynchronous inputs are active simultaneously, Reset overrides Set.

In addition to driving the chip output buffer, the Macrocell output is routed back as an input to the UIM. One private product term can be configured to control the Output Enable of the output buffer and/or the feedback to the UIM. If it is configured to control UIM feedback, the Output Enable product term forces the UIM feedback line High when the Macrocell output is disabled.

Universal Interconnect Matrix

The UIM receives inputs from each Macrocell output, I/O pin, and dedicated input pin. Acting as an unrestricted crossbar switch, the UIM generates 21 output signals to each High-Density Function Block and 24 output signals to each Fast Function Block.

Each UIM input can be programmed to connect to any UIM output. The delay through the interconnect matrix is constant, regardless of the routing distance, fan-out, or fan-in.

When multiple inputs are programmed to be connected to the same output, this output produces the logical AND of the input signals. By choosing the appropriate signal inversions at the input pins, Macrocell outputs and Function Block AND-array input, this AND logic can also be used to implement wide NAND, OR or NOR functions. This offers an additional level of logic without any speed penalty.

A Macrocell feedback signal that is disabled by the output enable product term represents a High input to the UIM. Programming several such Macrocell outputs onto the same UIM output emulates a 3-state bus line. If one of the Macrocell outputs is enabled, the UIM output assumes the enabled output's level.

Input/Output Blocks

Macrocells drive chip outputs directly through 3-state output buffers, each individually controlled by the Output Enable product term mentioned above. The Macrocell output can be inverted. An additional configuration option allows the output to be disabled permanently. Two dedicated FastOE inputs can also be configured to control any of the chip outputs instead of, or in conjunction with, the individual Output Enable product term. See Figure 7.

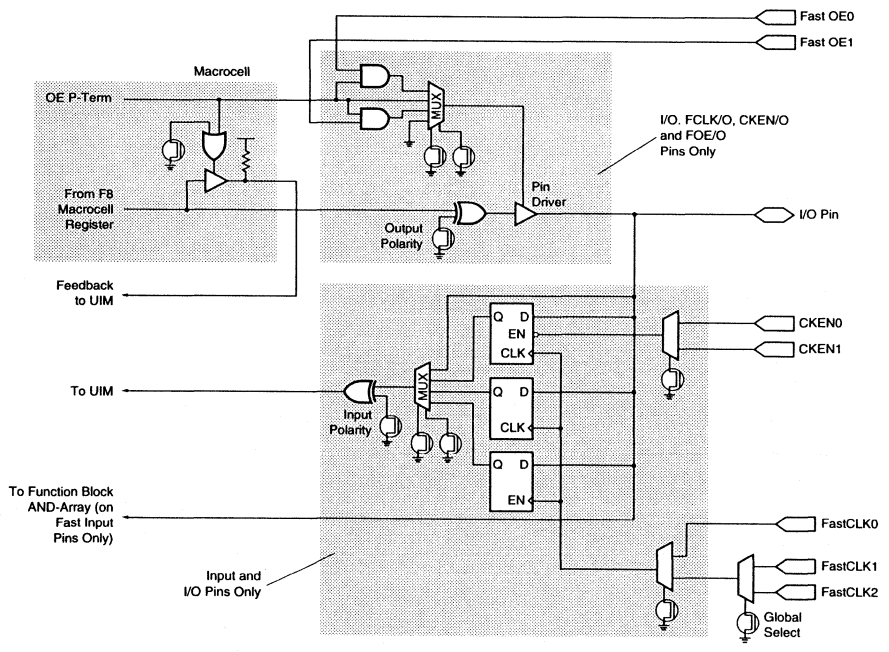


Figure 7. Input/Output Schematic

Each signal input to the chip is connected to a programmable input structure that can be configured as direct, latched, or registered. The latch and flip-flop can use one of two FastCLK signals as latch enable or clock. The two FastCLK signals are FCLK0 and a global choice of either FCLK1 or FCLK2. Latches are transparent when FastCLK is High, and flip-flops clock on the rising edge of FastCLK. The flip-flop includes an active-low clock enable, which when High, holds the present state of the flip-flop and inhibits response to the input signal. The clock enable source is one of two global Clock Enable signals ($\overline{CE0}$ and $\overline{CE1}$). An additional configuration option is polarity inversion for each input signal.

3.3 V or 5 V Interface Configuration

XC7300 devices can be used in systems with two different supply voltages: 3.3 V and 5 V. Each XC7300 device has separate V_{CC} connections to the internal logic and input buffers (V_{CCINT}) and to the I/O drivers (V_{CCIO}). V_{CCINT} must always be connected to a nominal 5 V supply, while V_{CCIO} may be connected to either 3.3 V or 5 V, depending on the output interface requirement.

When V_{CCIO} is connected to 5 V, the input thresholds are TTL levels, and thus compatible with 3.3 V and 5 V logic. The output High levels are also TTL compatible. When V_{CCIO} is connected to 3.3 V, the input thresholds are still TTL levels, and the outputs pull up to the 3.3 V rail. This makes the XC7300 ideal for interfacing directly to 3.3 V components. In addition, the output structure is designed so that the I/O can also safely interface to a mixed 3.3 V and 5 V bus.

Power-On Characteristics

Like many highly-flexible EPLDs, the XC7300 devices undergo a short internal initialization sequence upon device powerup. During this time, the outputs remain 3-stated while the device is configured from its internal EPROM array pattern and all registers are initialized. Except for the short delay during device initialization, this operation is completely transparent to the user and typically lasts 300 µs.

For additional flexibility, an active-Low Master Reset pin is provided so that EPLD can be reinitialized even after power is applied. It allows the EPLD to be initialized along with other devices in the system. When it is switched Low, all outputs become 3-stated and the initialization sequence is started. When it returns to High, the outputs become enabled and the device is ready for operation. If this flexibility is not needed, simply connect the Master Reset pin to the device V_{CCINT}.

During the initialization sequence, all input registers or latches are preloaded High and all FB and FFB Macrocell registers are preloaded to a known state. For FFB Macrocell registers where the Set/Reset product-term is defined, the preload is accomplished by asserting the product-term shortly before the end of the initialization sequence. When the Set/Reset product-term is defined and configured as Reset, the register preload value is Low. When the Set/Reset product-term is defined and configured as a Set, the register preload value is High. For FFB Macrocell registers where the Set/Reset product-term is not used, the register preload value is High.

For FB Macrocell registers, the preload value is defined by a separate preload configuration bit, independent of the Set and Reset product-terms. The value of this preload configuration bit is determined by the schematic capture library or in the user's design. If not specified, the register preload value is Low.

Power Management

The XC7300 family of devices feature a power-management scheme which permits non-speed-critical paths of a design to be operated at reduced power. Overall power dissipation is often reduced significantly, since, in most systems only a small part is speed critical.

Macrocells can individually be specified for high performance or low power operation by adding attributes to the logic schematic, or declaration statements to the behavioral description. To further conserve power, unused Macrocells are automatically turned off.

Erasure Characteristics

In windowed packages, the content of the EPROM array can be erased by exposure to ultraviolet light of wavelengths of approximately 4000 Å. The recommended erasure time is approximately 1 hr. when the device is placed within 1 in. of an ultraviolet lamp with a 12,000 $\mu W/cm^2$ power rating. To prevent unintentional exposure, place opaque labels over the device window.

When the device is exposed to high intensity UV light for much longer periods, permanent damage can occur. The maximum integrated dose the XC7300 EPLD can be exposed to without damage is 7000 W • s/cm^2, or approximately one week at 12,000 $\mu W/cm^2$.

Design Recommendations

For proper operation, all unused input and I/O pins must be connected to a valid logic level (High or Low). The recommended decoupling for all V_{CC} pins should total 1 µF using high-speed (tantalum or ceramic) capacitors.

Use electrostatic discharge (ESD) handling procedures with the XC7300-series EPLDs to prevent damage to the device during programming, assembly, and test.

Design Security

Each member of the XC7300 family has a multibit security system that controls access to the configuration programmed into the device. This security scheme uses multiple EPROM bits at various locations within the EPROM array to offer a higher degree of design security than other EPROM and fused-based devices. Programmed data within EPROM cells is invisible–even when examined under a microscope–and cannot be selectively erased. The EPROM security bits, and the device configuration data, reset when the device is erased.

High-Volume Production Programming

The XC7300 family offers flexibility for low-volume prototypes as well as cost-effectiveness for high-volume production. The designer can start with ceramic window package parts for prototypes, ramp up initial production using low-cost plastic parts programmed in-house, and then shift into high-volume production using Xilinx factory programmed and tested devices.

The Xilinx factory programmed concept offers significant advantages over competitive masked PLDs, or ASIC redesigns:

- No redesign is required – Even though masked devices are advertised as timing compatible, subtle differences in a chip layout can mean system failure.
- Devices are factory tested – Factory-programmed devices are tested as part of the manufacturing flow, insuring high-quality products.
- Shipments are delivered fast – Production shipments can begin within a few weeks, eliminating masking delays and qualification requirements.

For factory programming procedures, contact your local Xilinx representative.

Timing Model

Timing within the XC7300 EPLDs is accurately determined using external timing parameters from the device data sheet, using a variety of CAE simulators, or with the timing model shown in Figure 8.

The timing model is based on the fixed internal delays of the XC7300 architecture which consists of four basic parts: I/O Blocks, the UIM, Fast Function Blocks and High-Density Function Blocks. The timing model identifies the internal delay paths and their relationships to ac characteristics. Using this model and the ac characteristics, designers can easily calculate the timing information for a particular EPLD.

XEPLD Development System

The designer can create, implement, and verify digital logic circuits for EPLD devices using the Xilinx XEPLD Development System. Designs can be represented as schematics consisting of XEPLD library components, as behavioral descriptions, or as a mixture of both. The XEPLD translator maps the design quickly and automatically onto a chosen EPLD device, produces documentation for design analysis and creates a programming file to configure the device.

The following lists some of the XEPLD Development System features.

- Familiar design approach similar to TTL and PLD techniques
- Converts netlist to fuse map in minutes using a '486 PC or workstation platform
- Interfaces to standard third-party CAE schematics, simulation tools, and behavioral languages
- Schematic library with familiar and powerful TTL-like components, including PLDs and ALUs
- Predictable timing even before design entry, using library components and Boolean equations
- Timing simulation using Viewsim, OrCAD VST, and other tools controlled by the Xilinx Design Manager (XDM) program

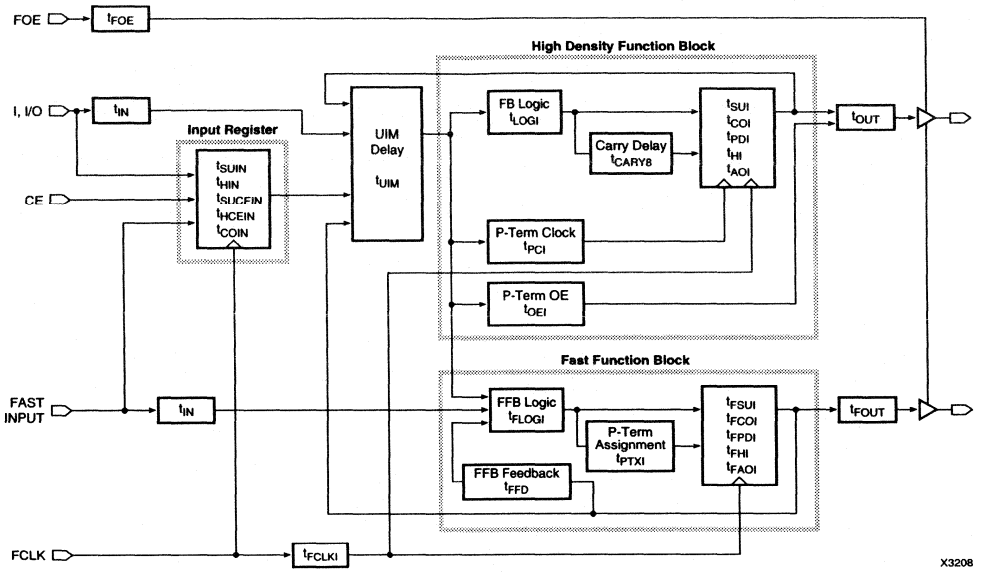
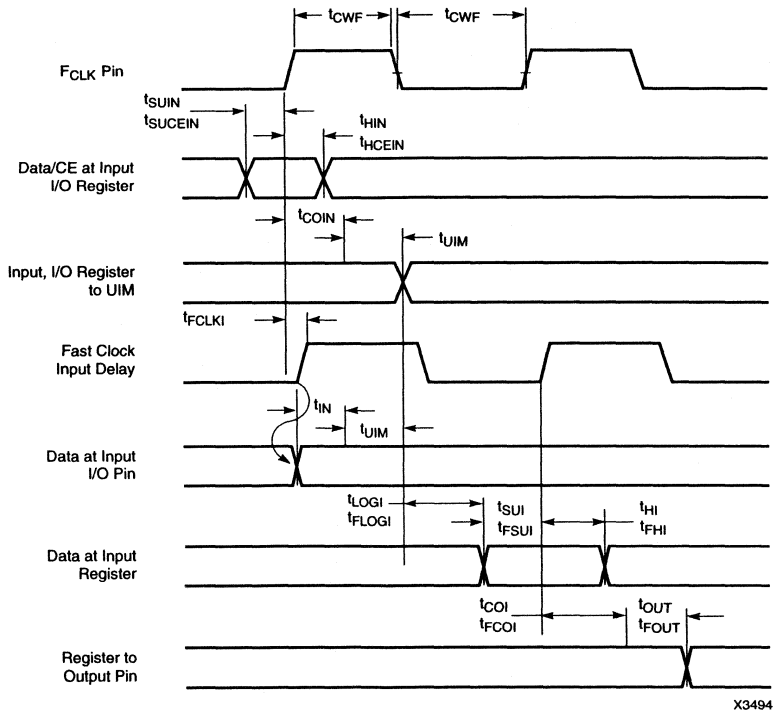
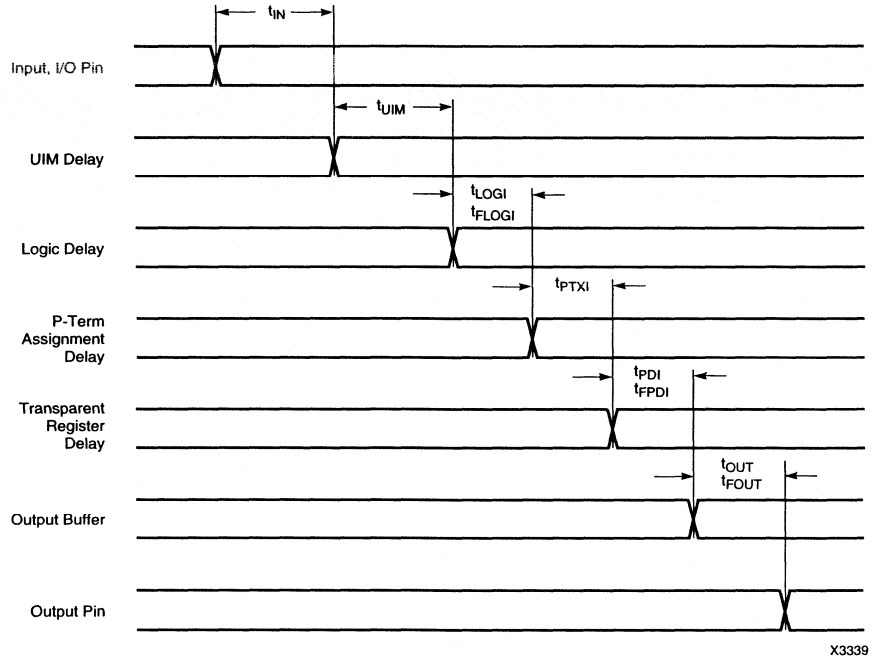


Figure 8. XC7300 Timing Model

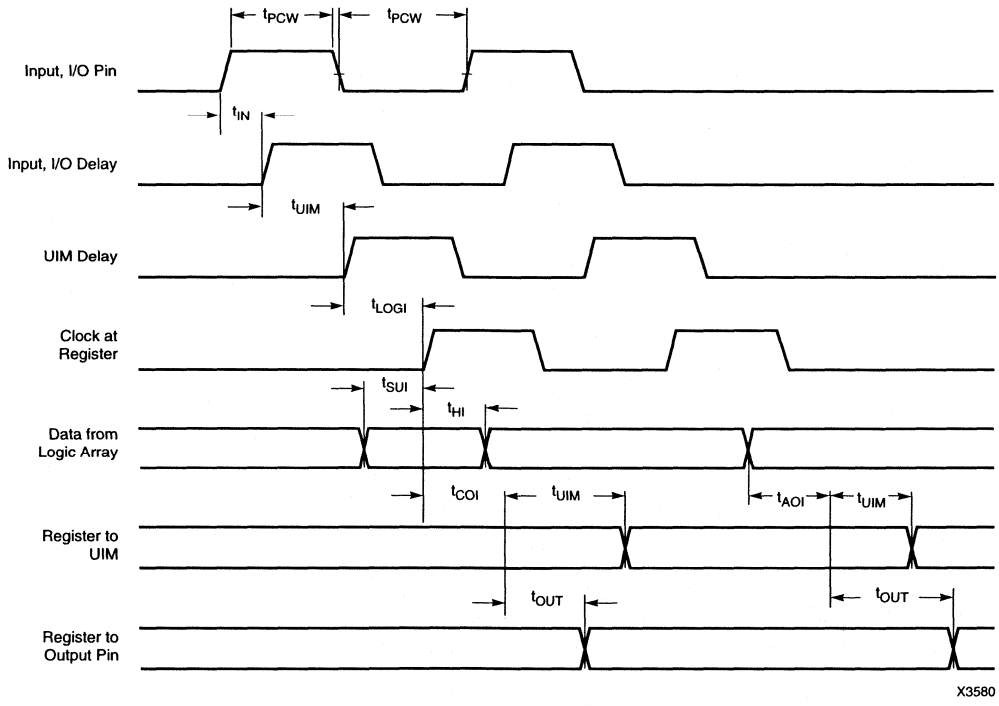
Synchronous Clock Switching Characteristics



Combinatorial Switching Characteristics



Asynchronous Clock Switching Characteristics



XC7336
36-Macrocell CMOS EPLD

Preliminary Product Specifications

Features

- Ultra high-performance EPLD
  - 7.5 ns pin-to-pin delay
  - 125 MHz maximum clock frequency
- Incorporates four Fast Function Blocks
- 100% interconnect matrix
- 36 Macrocells with programmable I/O architecture
- 36 outputs with 24 mA drive
- 3.3 V or 5 V I/O operation
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- Power management options
- Multiple security bits for design protection
- 44-pin leaded chip carrier package

General Description

The XC7336 is a member of the Xilinx Dual-Block EPLD family. It consists of four Fast Function Blocks interconnected by a central Universal Interconnect Matrix (UIM).

The four Function Blocks in the XC7336 (Figure 1) are PAL-like structures, complete with programmable product term arrays and programmable multilevel Macrocells. Each Function Block receives 24 inputs, contains nine Macrocells configurable for registered or combinatorial logic and produces nine outputs which feedback to the UIM and output pins.

The Universal Interconnect Matrix connects the Function Blocks to each other and to all input pins, providing 100% connectivity between the Function Blocks. This allows logic functions to be mapped into the Function Blocks and interconnected without routing restrictions.

The XC7336 device is designed in 0.8µ CMOS EPROM technology.

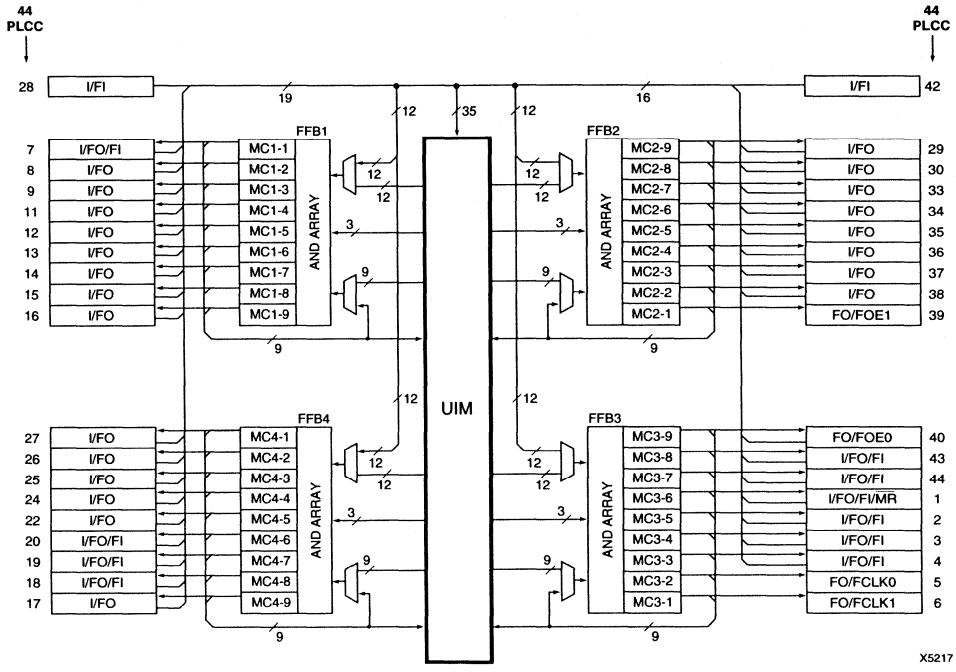


Figure 1. XC7336 Functional Block Diagram

In addition, the XC7336 includes a programmable power management feature to specify high-performance or low-power operation on an individual Macrocell-by-Macrocell basis. Unused Macrocells are automatically turned off to minimize power dissipation. Designers can operate speed-critical paths at maximum performance, while non-critical paths dissipate less power.

Xilinx development software (XEPLD) supports all members of XC7300 family. The designer can create, implement, and verify digital logic circuits for EPLD devices using the Xilinx XEPLD Development System. Designs can be represented as schematics consisting of XEPLD library components, as behavioral descriptions, or as a mixture of both. The XEPLD translator automatically performs logic optimization, collapsing, mapping and routing without user intervention. After compiling the design, XEPLD translator produces documentation for design analysis and creates a programming file to configure the device.

The following lists some of the XEPLD Development System features.

- Familiar design approach similar to TTL and PLD techniques
- Converts netlist to fuse map in minutes using a 386/486 PC or workstation platform
- Interfaces to standard third-party CAE schematics, simulation tools, and behavioral languages
- Timing simulation using Viewsim, OrCAD VST, Mentor, LMC and other tools compatible with the Xilinx Netlist Format (XNF)

The XC7336 device is available in plastic and ceramic laeaded chip carriers. Package options include both windowed ceramic for design prototypes and one-time programmable plastic versions for cost-effective production volume.

Power Management

The XC7336 power management scheme allows designers to control on-chip power dissipation by configuring individual Macrocells to operate in high-performance or low-power modes of operation. Unused Macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

$$I_{CC}\ (mA) = MC_{HP}\ (4.3) + MC_{LP}\ (2.6) + MC\ (0.005\ mA/MHz)\ f$$

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of Macrocells used

f = Clock frequency (MHz)

Figure 2 shows a typical operating current for the XC7336 programmed as two 16-bit counters and operating at the indicated clock frequency.

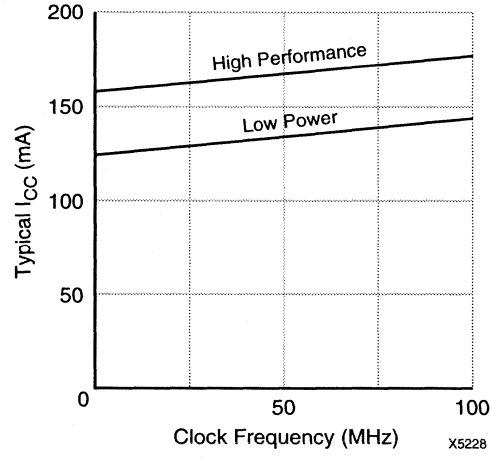


Figure 2. Typical I_{CC} vs Frequency for XC7336

Notice: The information contained in this data sheet pertains to products in the initial production phases of development. These specifications are subject to change without notice. Verify with your local Xilinx sales office that you have the latest sheet before finalizing a design.

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage with respect to GND	-0.5 to 7.0	V
V_{IN}	DC Input voltage with respect to GND	-0.5 to 7.0	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to 7.0	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Maximum soldering temperature (10s @ 1/16 in. = 1.5 mm)	+260	°C

***Warning**: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.*

Recommended Operating Conditions

Symbol	Parameter			Min	Max	Units
V_{CCINT}/ V_{CCIO}	Supply voltage relative to GND @ 5 V	Commercial	T_A = 0° C to70° C	4.75	5.25	V
	Supply voltage relative to GND @ 5 V	Industrial	T_A = -40° C to 85° C	4.50	5.50	V
	Supply voltage relative to GND @ 5 V	Military	T_C = -55° C to 125° C	4.50	5.50	V
V_{CCIO}	I/O supply voltage relative to GND @ 3.3 V			3.00	3.60	V
V_{IL}	Low-level input voltage			0	0.80	V
V_{IH}	High-level input voltage			2.00	V_{CC}+0.3	V
V_O	Output voltage			0	V_{CCIO}	V

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V Low-level output voltage	I_{OL} = 24 mA V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_O = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		6.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		8.0	pF
C_{OUT}(1)	Output capacitance	V_O = GND f = 1.0 MHz		10.0	pF
I_{CC1}(2)	Supply Current (low power mode)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5 V f = 1.0 MHz @ 25°C	126 mA Typ		

Notes: 1. Sample tested
2. Measured with device programmed as two 16-bit counters

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time (to outputs operational) following assertion of Master Reset	100		200	µs

Fast Function Block (FFB) External AC Characteristics[1]

Symbol	Parameter	XC7336-7 (Com only)		XC7336-10 (Com/Ind only)		XC7336-12		XC7336-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{CF}	Max count frequency [2,3]	125.0		100.0		80.0		66.7		MHz
t_{SUF}	Fast input setup time before FCLK ↑ [2]	4.0		5.0		6.0		7.0		ns
t_{HF}	Fast input hold time after FCLK ↑	0		0		0		0		ns
t_{COF}	FCLK ↑ to output valid		5.5		8.0		9.0		12.0	ns
t_{PDFO}	Fast input to output valid [2,3]		7.5		10.0		12.0		15.0	ns
t_{PDFU}	I/O to output valid [2,3]		12.0		15.0		19.0		23.0	ns
t_{CWF}	Fast clock pulse width	4.0		5.0		5.5		6.0		ns

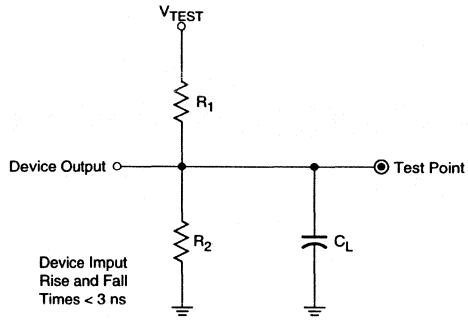
Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC7336-7 (Com only)		XC7336-10 (Com/Ind only)		XC7336-12		XC7336-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay [3]		1.5		1.5		2.0		2.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay [3]		3.5		5.5		7.0		8.0	ns
t_{FSUI}	FFB register setup time	1.5		2.5		3.0		4.0		ns
t_{FHI}	FFB register hold time	2.5		2.5		3.0		3.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		0.5		1.0		1.0	ns
t_{FAOI}	FFB register async. set delay		2.0		2.5		3.0		4.0	ns
t_{PTXI}	FFB p-term assignment delay		0.8		1.0		1.2		1.5	ns
t_{FFD}	FFB feedback delay		4.0		5.0		6.5		8.0	ns

Notes:
1. All appropriate ac specifications tested using Figure 3 as test load circuit
2. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$.
3. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.

Internal AC Characteristics

Symbol	Parameter	XC7336-7 (Com only)		XC7336-10 (Com/Ind only)		XC7336-12		XC7336-15		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		2.5		3.5		4.0		5.0	ns
t_{FOUT}	FFB output buffer and pad delay		3.0		4.5		5.0		7.0	ns
t_{UIM}	Universal Interconnect Matrix delay		4.5		5.0		7.0		8.0	ns
t_{FOE}	Fast output enable/disable buffer delay		7.5		10.0		12.0		15.0	ns
t_{FCLKI}	Fast clock buffer delay		1.5		2.5		3.0		4.0	ns



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
FO	5.0 V	5.0 V	160 Ω	120 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X3491

Figure 3. AC Load Circuit

XC7336 Pinouts

44 LCC	Pin Description
1	I/FO/FI/ $\overline{MR}$
2	I/FO/FI
3	I/FO/FI
4	I/FO/FI
5	FO/FCLK0
6	FO/FCLK1
7	I/FO/FI
8	I/FO
9	I/FO
10	GND
11	I/FO
12	I/FO
13	I/FO
14	I/FO
15	I/FO
16	I/FO
17	I/FO
18	I/FO/FI
19	I/FO/FI
20	I/FO/FI
21	V_{CCINT}
22	I/FO

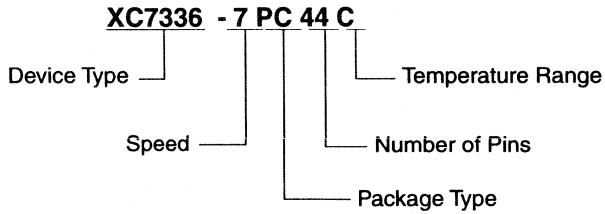
44 LCC	Pin Description
23	GND
24	I/FO
25	I/FO
26	I/FO
27	I/FO
28	I/FI
29	I/FO
30	I/FO
31	GND
32	V_{CCIO}
33	I/FO
34	I/FO
35	I/FO
36	I/FO
37	I/FO
38	I/FO
39	FO/FOE1
40	FO/FOE0
41	V_{CCINT}
42	I/FI
43	I/FO/FI
44	I/FO/FI

For a detailed description of the device architecture, see the XC7300 CMOS EPLD Family data sheet, page 3-5 through 3-11.

For a detailed description of the device timing, see pages 3-13, 3-14 and 3-19.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Speed Options

-15	15 ns pin-to-pin delay
-12	12 ns pin-to-pin delay
-10	10 ns pin-to-pin delay (commercial and industrial only)
-7	7.5 ns pin-to-pin delay (commercial only)

Packaging Options

PC44	44-Pin Plastic Leaded Chip Carrier
WC44	44-Pin Windowed Ceramic Leaded Chip Carrier

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	-40°C to 85°C
M	Military	-55°C to 125°C (Case)

Component Availability

Pins		44		68		84			144	160	184	225	
Type		Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Ceramic PGA	Ceramic PGA	Plastic PQFP	Ceramic PGA	Plastic BGA	Ceramic BGA
Code		PC44	WC44	PC68	WC68	PC84	WC84	PG84	PG144	PQ160	PG184	PB225	WB225
XC7336	-15	CI	CI										
	-12	CI	CI										
	-10	CI	CI										
	-7	C	C										

C = Commercial = 0° to +70°C I = Industrial = -40° to 85°C

X5276

XC7354
54-Macrocell CMOS EPLD

Preliminary Product Specifications

Features

- High-Performance EPLD
  - 10 ns pin-to-pin delay
  - 100 MHz maximum clock frequency
- Advanced Dual-Block architecture
  - Two Fast Function Blocks
  - Four High-Density Function Blocks
- 100% interconnect matrix
- High-Speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 48 MHz 18-bit accumulators
- 54 Macrocells with programmable I/O architecture
- Up to 54 inputs programmable as direct, latched, or registered
- 18 outputs with 24 mA drive
- 3.3 V or 5 V I/O operation
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- Power management options
- Multiple security bits for design protection
- 44- and 68-pin leaded chip carrier package

General Description

The XC7354 is a member of the Xilinx Dual-Block EPLD family. It consists of two Fast Function Blocks and fourHigh-Density Function Blocks interconnected by a central Universal Interconnect Matrix (UIM).

The six Function Blocks in the XC7354 (Figure 1) are PAL-like structures, complete with programmable product term arrays and programmable multilevel Macrocells. Each Function Block receives 24 inputs, contains nine Macrocells configurable for registered or combinatorial logic and produces nine outputs which feedback to the UIM.

The Universal Interconnect Matrix connects the Function Blocks to each other and to all input pins, providing 100% connectivity between the Function Blocks. This allows logic functions to be mapped into the Function Blocks and interconnected without routing restrictions.

Xilinx development software (XEPLD) supports all members of XC7300 family. The designer can create, implement, and verify digital logic circuits for EPLD devices using the Xilinx XEPLD Development System. Designs can be represented as schematics consisting of XEPLD library components, as behavioral descriptions, or as a mixture of both. The XEPLD translator automatically performs logic optimization, collapsing, mapping and routing without user intervention. After compiling the design, XEPLD translator produces documentation for design analysis and creates a programming file to configure the device.

The following lists some of the XEPLD Development System features.

- Familiar design approach similar to TTL and PLD techniques
- Converts netlist to fuse map in minutes using a 386/486 PC or workstation platform
- Interfaces to standard third-party CAE schematics, simulation tools, and behavioral languages
- Timing simulation using Viewsim, OrCAD VST, Mentor, LMC and other tools compatible with the Xilinx Netlist Format (XNF)

The XC7354 device is available in plastic and ceramic leaded chip carriers. Package options include both windowed ceramic for design prototypes and one-time programmable plastic versions for cost-effective production volume.

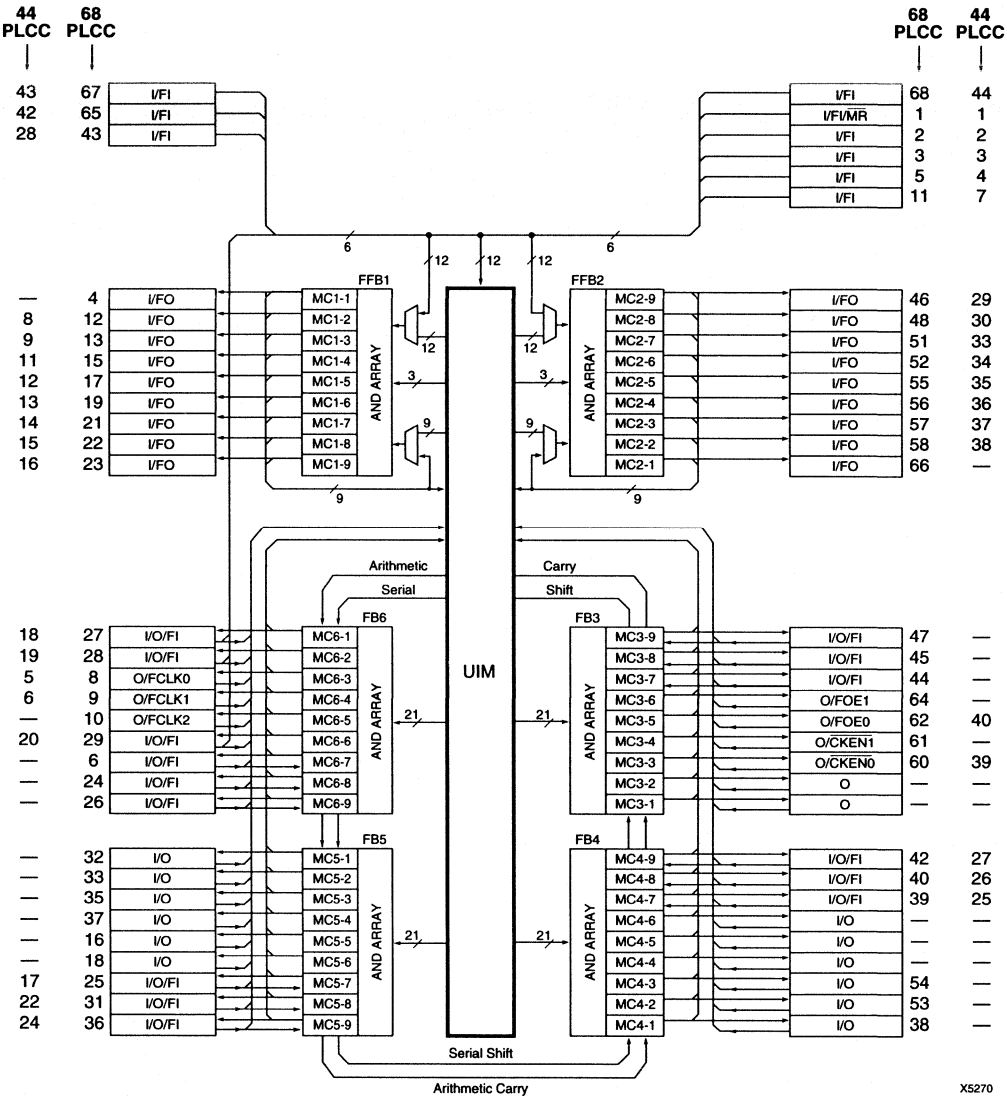


Figure 1. XC7354 Functional Block Diagram

Power Management

The XC7354 power management scheme allows designers to control on-chip power dissipation by configuring individual Macrocells to operate in high-performance or low-power modes of operation. Unused Macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

$$I_{CC}\ (mA) = MC_{HP}\ (3.0) + MC_{LP}\ (2.6) + MC\ (0.006\ mA/MHz)\ f$$

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of Macrocells used

f = Clock frequency (MHz)

Figure 2 shows a typical power calculation for the XC7354 device, programmed as three 16-bit counters and operating at the indicated clock frequency.

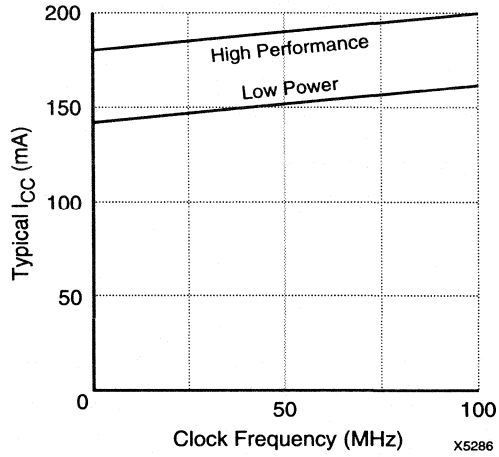


Figure 2. Typical I_{CC} vs Frequency for XC7354

Notice: The information contained in this data sheet pertains to products in the initial production phases of development. These specifications are subject to change without notice. Verify with your local Xilinx sales office that you have the latest data sheet before finalizing a design.

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage with respect to GND	-0.5 to 7.0	V
V_{IN}	DC Input voltage with respect to GND	-0.5 to 7.0	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to 7.0	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in. = 1.5 mm)	+260	°C

Warning: *Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.*

Recommended Operating Conditions

Symbol	Parameter			Min	Max	Units
V_{CCINT}/ V_{CCIO}	Supply voltage relative to GND @ 5 V	Commercial	T_A = 0°C to 70°C		5.25	V
	Supply voltage relative to GND @ 5 V	Industrial	T_A = -40° C to 85° C	4.5	5.5	V
	Supply voltage relative to GND @ 5 V	Military	T_C = -55° C to 125° C	4.5	5.5	V
V_{CCIO}	I/O supply voltage relative to GND @ 3.3 V			3.0	3.6	V
V_{IL}	Low-level input voltage			0	0.8	V
V_{IH}	High-level input voltage			2.0	V_{CC}+0.3	V
V_O	Output voltage			0	V_{CCIO}	V

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V Low-level output voltage	I_{OL} = 24 mA I_{OL} = 12 mA V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 10 mA (I/O) V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_O = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		10.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FCLK2, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		15.0	pF
C_{OUT}[1]	Output capacitance	V_O = GND f = 1.0 MHz		20.0	pF
I_{CC1}[2]	Supply Current (low power mode)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5 V f = 1.0 MHz @ 25°C	140 mA Typ		

Noes: 1. Sample tested
2. Measured with device programmed as three 16-bit counters

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time (to outputs operational) following assertion of Master Reset	200		300	µs

Fast Function Block (FFB) External AC Characteristics[3]

Symbol	Parameter	XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	
f_{CF}	Max count frequency (1, 2)	100.0		80.0		66.7		MHz
t_{SUF}	Fast input setup time before FCLK ↑ (1)	5.0		6.0		7.0		ns
t_{HF}	Fast input hold time after FCLK ↑	0		0		0		ns
t_{COF}	FCLK ↑ to output valid		8.0		9.0		12.0	ns
t_{PDFO}	Fast input to output valid (1, 2)		10.0		12.0		15.0	ns
t_{PDFU}	I/O to output valid (1, 2)		16.0		19.0		23.0	ns
t_{CWF}	Fast clock pulse width	5.0		5.5		6.0		ns

High-Density Function Block (FB) External AC Characteristics

Symbol	Parameter	XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	
f_C	Max count frequency (1, 2)	76.9		66.7		55.6		MHz
t_{SU}	I/O setup time before FCLK ↑ (1, 2)	13.0		15.0		18.0		ns
t_H	I/O hold time after FCLK ↑	0		0		0		ns
t_{CO}	FCLK ↑ to output valid		10.0		12.0		15.0	ns
t_{PSU}	I/O setup time before p-term clock ↑ (2)	6.0		7.0		9.0		ns
t_{PH}	I/O hold time after p-term clock ↑	0		0		0		ns
t_{PCO}	P-term clock ↑ to output valid		17.0		20.0		24.0	ns
t_{PD}	I/O to output valid (1, 2)		22.0		27.0		32.0	ns
t_{CW}	Fast clock pulse width	5.0		5.5		6.0		ns
t_{PCW}	P-term clock pulse width	6.0		7.5		8.5		ns

Notes:
1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.
3. AC test load used for all parameters except where noted.

Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay (2)		1.5		2.0		2.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay (2)		5.5		7.0		8.0	ns
t_{FSUI}	FFB register setup time	2.5		3.0		4.0		ns
t_{FHI}	FFB register hold time	2.5		3.0		3.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		1.0		1.0	ns
t_{FAOI}	FFB register async. set delay		2.5		3.0		4.0	ns
t_{PTXI}	FFB p-term assignment delay		1.0		1.2		1.5	ns
t_{FFD}	FFB feedback delay		5.0		6.5		8.0	ns

High-Density Function Block (FB) Internal AC Characteristics

Symbol	Parameter	XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	
t_{LOGI}	FB logic array delay (2)		3.5		4.0		5.0	ns
t_{LOGILP}	Low power FB logic delay (2)		7.5		9.0		11.0	ns
t_{SUI}	FB register setup time	2.5		3.0		4.0		ns
t_{HI}	FB register hold time	3.5		4.0		5.0		ns
t_{COI}	FB register clock-to-output delay		1.0		1.0		1.0	ns
t_{PDI}	FB register pass through delay		2.5		4.0		4.0	ns
t_{AOI}	FB register async. set/reset delay		3.0		4.0		5.0	ns
t_{RA}	Set/reset recovery time before FCLK ↑	16.0		18.0		21.0		ns
t_{HA}	Set/reset hold time after FCLK ↑	0		0		0		ns
t_{PRA}	Set/reset recovery time before p-term clock ↑	10.0		12.0		15.0		ns
t_{PHA}	Set/reset hold time after p-term clock ↑	6.0		8.0		9.0		ns
t_{PCI}	FB p-term clock delay		0		0		0	ns
t_{OEI}	FB p-term output enable delay		4.0		5.0		7.0	ns
t_{CARY8}	ALU carry delay within 1 FB (3)		6.0		8.0		12.0	ns
t_{CARYFB}	Carry lookahead delay per additional Functional Block (3)		1.5		2.0		3.0	ns

Notes: 2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.
3. Arithmetic carry delays are measured as the increase in required set-up time to adjacent Macrocell(s) for adder with registered outputs.

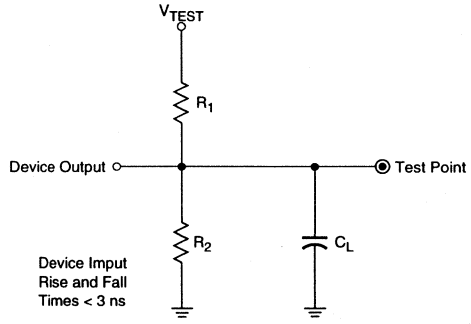
I/O Block External AC Characteristics

Symbol	Parameter	XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	
f_{IN}	Max pipeline frequency (input register to FFB or FB register) (2)	76.9		66.7		55.6		MHz
t_{SUIN}	Input register/latch setup time before FCLK ↑	5.0		6.0		7.0		ns
t_{HIN}	Input register/latch hold time after FCLK ↑	0		0		0		ns
t_{COIN}	FCLK ↑ to input register/latch output		3.5		4.0		5.0	ns
t_{CESUIN}	Clock enable setup time before FCLK ↑	7.0		8.0		10.0		ns
t_{CEHIN}	Clock enable hold time after FCLK ↑	0		0		0		ns
t_{CWHIN}	FCLK pulse width high time	5.0		5.5		6.0		ns
t_{CWLIN}	FCLK pulse width low time	5.0		5.5		6.0		ns

Internal AC Characteristics

Symbol	Parameter	XC7354-10 (Com/Ind only)		XC7354-12		XC7354-15		Units
		Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		3.5		4.0		5.0	ns
t_{FOUT}	FFB output buffer and pad delay		4.5		5.0		7.0	ns
t_{OUT}	FB output buffer and pad delay		6.5		8.0		10.0	ns
t_{UIM}	Universal Interconnect Matrix delay		6.0		7.0		8.0	ns
t_{FOE}	Fast output enable/disable buffer delay		10.0		12.0		15.0	ns
t_{FCLKI}	Fast clock buffer delay		2.5		3.0		4.0	ns

Note: 2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
FO	5.0 V	5.0 V	160 Ω	120 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF
O	5.0 V	5.0 V	310 Ω	195 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X3491

Figure 3. AC Load Circuit

XC7354 Pinouts

68 LCC	44 LCC	Pin Description
1	1	I/FI/ $\overline{MR}$
2	2	I/FI
3	3	I/FI
4	–	I/FO
5	4	I/FI
6	–	I/O/FI
7	–	GND
8	5	O/FCLK0
9	6	O/FCLK1
10	–	O/FCLK2
11	7	I/FI
12	8	I/FO
13	9	I/FO
14	10	GND
15	11	I/FO
16	–	I/O
17	12	I/FO
18	–	I/O
19	13	I/FO
20	–	V_{CCIO}
21	14	I/FO
22	15	I/FO
23	16	I/FO
24	–	I/O/FI
25	17	I/O/FI
26	–	I/O/FI
27	18	I/O/FI
28	19	I/O/FI
29	20	I/O/FI
30	21	V_{CCINT}
31	22	I/O/FI
32	–	I/O
33	–	I/O
34	23	GND

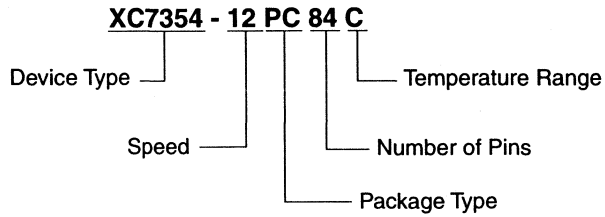
68 LCC	44 LCC	Pin Description
35	–	I/O
36	24	I/O/FI
37	–	I/O
38	–	I/O
39	25	I/O/FI
40	26	I/O/FI
41	–	GND
42	27	I/O/FI
43	28	I/FI
44	–	I/O/FI
45	–	I/O/FI
46	29	I/FO
47	–	I/O/FI
48	30	I/FO
49	31	GND
50	32	V_{CCIO}
51	33	I/FO
52	34	I/FO
53	–	I/O
54	–	I/O
55	35	I/FO
56	36	I/FO
57	37	I/FO
58	38	I/FO
59	–	V_{CCINT}
60	39	O/$\overline{CKEN0}$
61	–	O/$\overline{CKEN1}$
62	40	O/FOE0
63	41	V_{CCINT}/V_{PP}
64	–	O/FOE1
65	42	I/FI
66	–	I/FO
67	43	I/FI
68	44	I/FI

For a detailed description of the device architecture, see the XC7300 CMOS EPLD Family data sheet, page 3-5 through 3-11.

For a detailed description of the device timing, see pages 3-13, 3-14 and 3-28 through 3-30.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Speed Options

-15	15 ns pin-to-pin delay
-12	12 ns pin-to-pin delay
-10	10 ns pin-to-pin delay (commerial and industrial only)

Packaging Options

PC44	44-Pin Plastic Leaded Chip Carrier
WC44	44-Pin Windowed Ceramic Leaded Chip Carrier
PC68	68-Pin Plastic Leaded Chip Carrier
WC68	68-Pin Windowed Ceramic Leaded Chip Carrier

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	-40°C to 85°C
M	Military	-55°C to 125°C (Case)

Component Availability

Pins		44		68		84			144	160	184	225	
Type		Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Ceramic PGA	Ceramic PGA	Plastic PQFP	Ceramic PGA	Plastic BGA	Ceramic BGA
Code		PC44	WC44	PC68	WC68	PC84	WC84	PG84	PG144	PQ160	PG184	PB225	WB225
XC7354	-15	CI	CI	CI	CI								
	-12	CI	CI	CI	CI								
	-10	CI	CI	CI	CI								

C = Commercial = 0° to +70°C I = Industrial = -40° to 85°C

X5277

XC7372
72-Macrocell CMOS EPLD

Preliminary Product Specifications

Features

- High-Performance EPLD
  - 10 ns pin-to-pin delay
  - 100 MHz maximum clock frequency
- Advanced Dual-Block architecture
  - Two Fast Function Blocks
  - Six High-Density Function Blocks
- 100% interconnect matrix
- High-Speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 46 MHz 18-bit accumulators
- 72 Macrocells with programmable I/O architecture
- Up to 90 inputs programmable as direct, latched, or registered
- 18 outputs with 24 mA drive
- 3.3 V or 5 V I/O operation
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- Power management options
- Multiple security bits for design protection
- 68-, 84-pin leaded chip carrier and 84-pin Pin-Grid-Array packages

General Description

The XC7372 is a member of the Xilinx Dual-Block EPLD family. It consists of two Fast Function Blocks and six High-Density Function Blocks interconnected by a central Universal Interconnect Matrix (UIM).

The eight Function Blocks in the XC7372 (Figure 1) are PAL-like structures, complete with programmable product term arrays and programmable multilevel Macrocells. Each Function Block receives 24 inputs, contains nine Macrocells configurable for registered or combinatorial logic and produces nine outputs which feedback to the UIM.

The Universal Interconnect Matrix connects the Function Blocks to each other and to all input pins, providing 100% connectivity between the Function Blocks. This allows logic functions to be mapped into the Function Blocks and interconnected without routing restrictions.

The XC7372 device is designed in 0.8µ CMOS EPROM technology.

In addition, the XC7372 includes a programmable power management feature to specify high-performance or low-power operation on an individual Macrocell-by-Macrocell basis. Unused Macrocells are automatically turned off to minimize power dissipation. Designers can operate speed-critical paths at maximum performance, while non-critical paths dissipate less power.

Xilinx development software (XEPLD) supports all members of the XC7300 family. The designer can create, implement, and verify digital logic circuits for EPLD devices using the Xilinx XEPLD Development System. Designs can be represented as schematics consisting of XEPLD library components, as behavioral descriptions, or as a mixture of both. The XEPLD translator automatically performs logic optimization, collapsing, mapping and routing without user intervention. After compiling the design, XEPLD translator produces documentation for design analysis and creates a programming file to configure the device.

The following lists some of the XEPLD Development System features.

- Familiar design approach similar to TTL and PLD techniques
- Converts netlist to fuse map in minutes using a 386/486 PC or workstation platform
- Interfaces to standard third-party CAE schematics, simulation tools, and behavioral languages
- Timing simulation using Viewsim, OrCAD VST, Mentor, LMC and other tools compatible with the Xilinx Netlist Format (XNF)

The XC7372 device is available in plastic and ceramic leaded chip carriers and pin-grid-arrays. Package options include both windowed ceramic for design prototypes and one-time programmable plastic versions for cost-effective production volume.

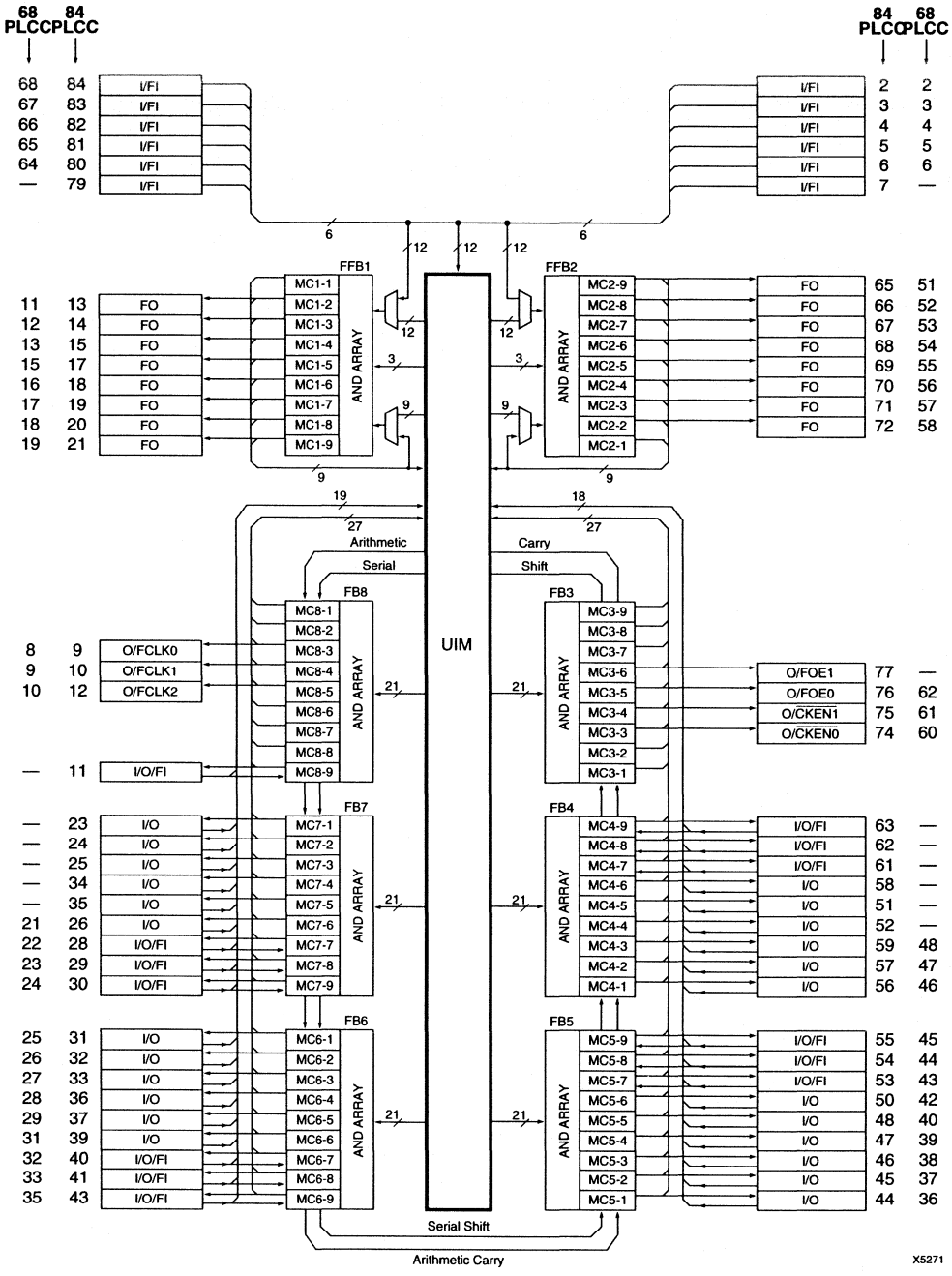


Figure 1. XC7372 Functional Block Diagram

Power Management

The XC7372 power management scheme allows designers to control on-chip power dissipation by configuring individual Macrocells to operate in high-performance or low-power modes of operation. Unused Macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

$$I_{CC}\ (mA) = MC_{HP}\ (3.1) + MC_{LP}\ (2.6) + MC\ (0.012\ mA/MHz)\ f$$

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of Macrocells used

f = Clock frequency (MHz)

Figure 2 shows typical power calculation for the XC7372 programmed as four 16-bit counters and operating at the indicated clock frequency.

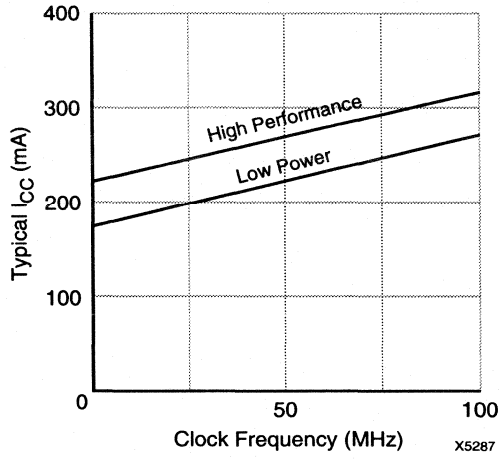


Figure 2. Typical I_{CC} vs Frequency for XC7372

Notice: The information contained in this data sheet pertains to products in the initial production phases of development. These specifications are subject to change without notice. Verify with your local Xilinx sales office that you have the latest data sheet before finalizing a design.

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage with respect to GND	-0.5 to 7.0	V
V_{IN}	DC Input voltage with respect to GND	-0.5 to 7.0	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to 7.0	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in. = 1.5 mm)	+260	°C

Warning: *Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.*

Recommended Operating Conditions

Symbol	Parameter		Min	Max	Units
V_{CCINT}/ V_{CCIO}	Supply voltage relative to GND @ 5 V	Commercial T_A=0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND @ 5 V	Industrial T_A=-40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND @ 5 V	Military T_C= -55°C to 125°C	4.5	5.5	V
V_{CCIO}	I/O supply voltage relative to GND @ 3.3 V		3.0	3.6	V
V_{IL}	Low-level input voltage		0	0.8	V
V_{IH}	High-level input voltage		2.0	V_{CC}+0.3	V
V_O	Output voltage		0	V_{CCIO}	V

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V Low-level output voltage	I_{OL} = 24 mA (I/O) I_{OL} = 12 mA, V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_O = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		10.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FCLK2, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		15.0	pF
C_{OUT}(1)	Output capacitance	V_O = GND f = 1.0 MHz		20.0	pF
I_{CC1}(2)	Supply Current (low power mode)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5 V f = 1.0 MHz @ 25°C	187 mA Typ		

Notes: 1. Sample tested
2. Measured with device programmed as four 16-bit counters

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time (to outputs operational) following assertion of Master Reset			300	µs

Fast Function Block (FFB) External AC Characteristics

Symbol	Parameter	XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	
f_{CF}	Max count frequency (1, 2)	100.0		80.0		66.7		MHz
t_{SUF}	Fast input setup time before FCLK ↑ (1)	5.0		6.0		7.0		ns
t_{HF}	Fast input hold time after FCLK ↑	0		0		0		ns
t_{COF}	FCLK ↑ to output valid		8.0		9.0		12.0	ns
t_{PDFO}	Fast input to output valid (1, 2)		10.0		12.0		15.0	ns
t_{PDFU}	I/O to output valid (1, 2)		17.0		20.0		24.0	ns
t_{CWF}	Fast clock pulse width	5.0		5.5		6.0		ns

High-Density Function Block (FB) External AC Characteristics

Symbol	Parameter	XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	
f_C	Max count frequency (1, 2)	71.4		62.5		52.6		MHz
t_{SU}	I/O setup time before FCLK ↑ (1, 2)	14.0		16.0		19.0		ns
t_H	I/O hold time after FCLK ↑	0		0		0		ns
t_{CO}	FCLK ↑ to output valid		10.0		12.0		15.0	ns
t_{PSU}	I/O setup time before p-term clock ↑ (2)	6.0		7.0		9.0		ns
t_{PH}	I/O hold time after p-term clock ↑	0		0		0		ns
t_{PCO}	P-term clock ↑ to output valid		18.0		21.0		25.0	ns
t_{PD}	I/O to output valid (1, 2)		23.0		28.0		33.0	ns
t_{CW}	Fast clock pulse width	5.0		5.5		6.0		ns
t_{PCW}	P-term clock pulse width	6.0		7.5		8.5		ns

Notes:
1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.
3. AC test load used for all parameters except where noted.

Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay (2)		1.5		2.0		2.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay (2)		5.5		7.0		8.0	ns
t_{FSUI}	FFB register setup time	2.5		3.0		4.0		ns
t_{FHI}	FFB register hold time	2.5		3.0		3.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		1.0		1.0	ns
t_{FAOI}	FFB register async. set delay		2.5		3.0		4.0	ns
t_{PTXI}	FFB p-term assignment delay		1.0		1.2		1.5	ns
t_{FFD}	FFB feeback delay		5.0		6.5		8.0	ns

High-Density Function Block (FB) Internal AC Characteristics

Symbol	Parameter	XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	
t_{LOGI}	FB logic array delay (2)		3.5		4.0		5.0	ns
t_{LOGILP}	Low power FB logic delay (2)		7.5		9.0		11.0	ns
t_{SUI}	FB register setup time	2.5		3.0		4.0		ns
t_{HI}	FB register hold time	3.5		4.0		5.0		ns
t_{COI}	FB register clock-to-output delay		1.0		1.0		1.0	ns
t_{PDI}	FB register pass through delay		2.5		4.0		4.0	ns
t_{AOI}	FB register async. set/reset delay		3.0		4.0		5.0	ns
t_{RA}	Set/reset recovery time before FCLK ↑	17.0		19.0		22.0		ns
t_{HA}	Set/reset hold time after FCLK ↑	0		0		0		ns
t_{PRA}	Set/reset recovery time before p-term clock ↑	10.0		12.0		15.0		ns
t_{PHA}	Set/reset hold time after p-term clock ↑	6.0		8.0		9.0		ns
t_{PCI}	FB p-term clock delay		0		0		0	ns
t_{OEI}	FB p-term output enable delay		4.0		5.0		7.0	ns
t_{CARY8}	ALU carry delay within 1 FB (3)		6.0		8.0		12.0	ns
t_{CARYFB}	Carry lookahead delay per additional Functional Block (3)		1.5		2.0		3.0	ns

Notes: 2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.
3. Arithmetic carry delays are measured as the increase in required set-up time to adjacent Macrocell(s) for adder with registered outputs.

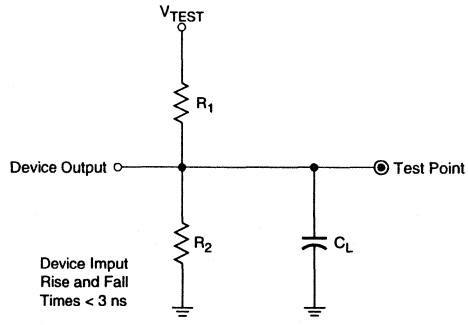
I/O Block External AC Characteristics

Symbol	Parameter	XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	
f_{IN}	Max pipeline frequency (input register to FFB or FB register) (2)	71.4		62.5		52.6		MHz
t_{SUIN}	Input register/latch setup time before FCLK ↑	5.0		6.0		7.0		ns
t_{HIN}	Input register/latch hold time after FCLK ↑	0		0		0		ns
t_{COIN}	FCLK ↑ to input register/latch output		3.5		4.0		5.0	ns
t_{CESUIN}	Clock enable setup time before FCLK ↑	7.0		8.0		10.0		ns
t_{CEHIN}	Clock enable hold time after FCLK ↑	0		0		0		ns
t_{CWHIN}	FCLK pulse width high time	5.0		5.5		6.0		ns
t_{CWLIN}	FCLK pulse width low time	5.0		5.5		6.0		ns

Internal AC Characteristics

Symbol	Parameter	XC7372-10 (Com/Ind only)		XC7372-12		XC7372-15		Units
		Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		3.5		4.0		5.0	ns
t_{FOUT}	FFB output buffer and pad delay		4.5		5.0		7.0	ns
t_{OUT}	FB output buffer and pad delay		6.5		8.0		10.0	ns
t_{UIM}	Universal Interconnect Matrix delay		7.0		8.0		9.0	ns
t_{FOEI}	Fast output enable/disable buffer delay		10.0		12.0		15.0	ns
t_{FCLKI}	Fast clock buffer delay		2.5		3.0		4.0	ns

Note: 2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
FO	5.0 V	5.0 V	160 Ω	120 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X3491

Figure 3. AC Load Circuit

84 LCC	68 LCC	Pin Description
1	1	$\overline{MR}$
2	2	I/FI
3	3	I/FI
4	4	I/FI
5	5	I/FI
6	6	I/FI
7	–	I/FI
8	7	**GND**
9	8	O/FCLK0
10	9	O/FCLK1
11	–	I/O/FI
12	10	O/FCLK2
13	11	FO
14	12	FO
15	13	FO
16	14	**GND**
17	15	FO
18	16	FO
19	17	FO
20	18	FO
21	19	FO
22	20	V_{CCIO}
23	–	I/O
24	–	I/O
25	–	I/O
26	21	I/ O
27	–	**GND**
28	22	I/O/FI
29	23	I/OFI
30	24	I/O/FI
31	25	I/O
32	26	I/O
33	27	I/O
34	–	I/O
35	–	I/O
36	28	I/O
37	29	I/O
38	30	V_{CCINT}
39	31	I/O
40	32	I/O/FI
41	33	I/O/FI
42	34	**GND**

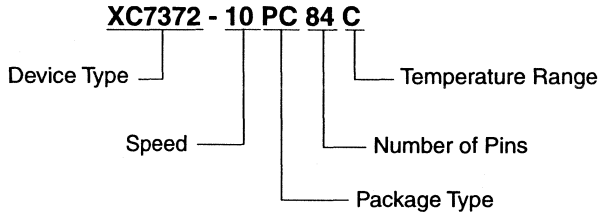
84 LCC	68 LCC	Pin Description
43	35	I/O/FI
44	36	I/O
45	37	I/O
46	38	I/O
47	39	I/O
48	40	I/O
49	41	**GND**
50	42	I/O
51	–	I/O
52	–	I/O
53	43	I/OFI
54	44	I/O/FI
55	45	I/O/FI
56	46	I/O
57	47	I/O
58	–	I/O
59	48	I/O
60	49	**GND**
61	–	I/O/FI
62	–	I/O/FI
63	–	I/O/FI
64	50	V_{CCIO}
65	51	FO
66	52	FO
67	53	FO
68	54	FO
69	55	FO
70	56	FO
71	57	FO
72	58	FO
73	59	V_{CCINT}
74	60	O/$\overline{CKEN0}$
75	61	O/$\overline{CKEN1}$
76	62	O/FOE0
77	–	O/FOE1
78	63	V_{CCINT}/V_{PP}
79	–	I/FI
80	64	I/FI
81	65	I/FI
82	66	I/FI
83	67	I/FI
84	68	I/FI

For a detailed description of the device architecture, see the XC7300 CMOS EPLD Family data sheet, page 3-5 through 3-11.

For a detailed description of the device timing, see pages 3-13, 3-14 and 3-38 through 3-40.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Speed Options

-15 15 ns pin-to-pin delay
-12 12 ns pin-to-pin delay
-10 10 ns pin-to-pin delay (commercial and industrial only)

Packaging Options

PC68 68-Pin Plastic Leaded Chip Carrier
WC68 68-Pin Windowed Ceramic Leaded Chip Carrier
PC84 84-Pin Plastic Leaded Chip Carrier
WC84 84-Pin Windowed Ceramic Leaded Chip Carrier
PG84 84-Pin Ceramic Pin Grid Array

Temperature Options

C Commercial 0°C to 70°C
I Industrial -40°C to 85°C
M Military -55°C to 125°C (Case)

Component Availability

Pins	44		68		84			144	160	184	225	
Type	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Ceramic PGA	Ceramic PGA	Plastic PQFP	Ceramic PGA	Plastic BGA	Ceramic BGA
Code	PC44	WC44	PC68	WC68	PC84	WC84	PG84	PG144	PQ160	PG184	PB225	WB225
-15			CI	CI	CI	CI	(CI)					
XC7372 -12			CI	CI	CI	CI	(CI)					
-10			CI	CI	CI	CI	(CI)					

C = Commercial = 0° to +70°C I = Industrial = -40° to 85°C
Parenthesis indicate future product plans

X5278

XC73108
108-Macrocell CMOS EPLD

Product Specifications

Features

- High-Performance EPLD
  - 12 ns pin-to-pin delay
  - 80 MHz maximum clock frequency
- Advanced Dual-Block architecture
  - 2 Fast Function Blocks
  - 10 High-Density Function Blocks
- 100% interconnect matrix
- High-Speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 42 MHz 18-bit accumulators
- 108 Macrocells with programmable I/O architecture
- Up to 90 inputs programmable as direct, latched, or registered
- 18 outputs with 24 mA drive
- 3.3 V or 5 V I/O operation
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- Power management options
- Multiple security bits for design protection
- 84-pin leaded chip carrier, 144-pin pin-grid-array packages, 160-pin plastic quad flat pack and 225-pin ball grid array

General Description

The XC73108 is a member of the Xilinx Dual-Block EPLD family. It consists of two Fast Function Blocks and ten High-Density Function Blocks interconnected by a central Universal Interconnect Matrix (UIM).

The eight Function Blocks in the XC73108 (Figure 1) are PAL-like structures, complete with programmable product term arrays and programmable multilevel Macrocells. Each Function Block receives 24 inputs, contains nine Macrocells configurable for registered or combinatorial logic and produces nine outputs which feedback to the UIM.

The Universal Interconnect Matrix connects the Function Blocks to each other and to all input pins, providing 100% connectivity between the Function Blocks. This allows logic functions to be mapped into the Function Blocks and interconnected without routing restrictions.

The XC73108 device is designed in 0.8µ CMOS EPROM technology.

In addition, the XC73108 includes a programmable power management feature to specify high-performance or low-power operation on an individual Macrocell-by-Macrocell basis. Unused Macrocells are automatically turned off to minimize power dissipation. Designers can operate speed-critical paths at maximum performance, while non-critical paths dissipate less power.

Xilinx development software (XEPLD) supports all members of the XC7300 family. The designer can create, implement, and verify digital logic circuits for EPLD devices using the Xilinx XEPLD Development System. Designs can be represented as schematics consisting of XEPLD library components, as behavioral descriptions, or as a mixture of both. The XEPLD translator automatically performs logic optimization, collapsing, mapping and routing without user intervention. After compiling the design, XEPLD translator produces documentation for design analysis and creates a programming file to configure the device.

The following lists some of the XEPLD Development System features.

- Familiar design approach similar to TTL and PLD techniques
- Converts netlist to fuse map in minutes using a 386/486 PC or workstation platform
- Interfaces to standard third-party CAE schematics, simulation tools, and behavioral languages
- Timing simulation using Viewsim, OrCAD VST, Mentor, LMC and other tools compatible with the Xilinx Netlist Format (XNF)

The XC73108 device is available in plastic and ceramic leaded chip carriers, pin-grid-arrays, plastic quad flat packs and ball-grid-array packages. Package options include both windowed ceramic for design prototypes and one-time programmable plastic versions for cost-effective production volume.

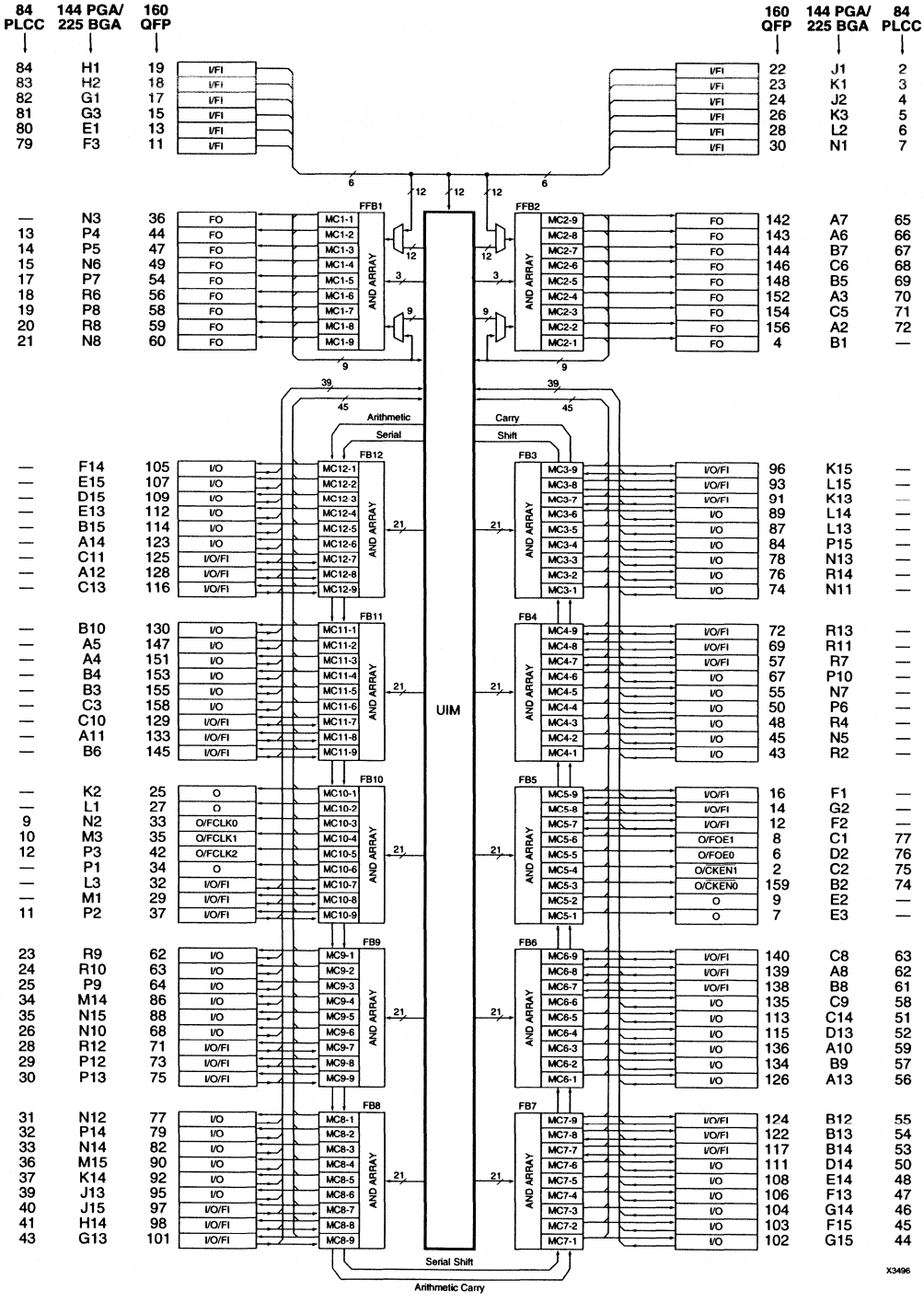


Figure 1. XC73108 Functional Block Diagram

Power Management

The XC73108 power management scheme allows designers to control on-chip power dissipation by configuring individual Macrocells to operate in high-performance or low-power modes of operation. Unused Macrocells are turned off to minimize power dissipation.

Operating current for each design can be approximated for specific operating conditions using the following equation:

$$I_{CC}\ (mA) = MC_{HP}\ (2.4) + MC_{LP}\ (2.1) + MC\ (0.015\ mA/MHz)\ f$$

Where:

MC_{HP} = Macrocells in high-performance mode

MC_{LP} = Macrocells in low-power mode

MC = Total number of Macrocells used

f = Clock frequency (MHz)

Figure 2 shows a typical calculation for the XC73108 device.

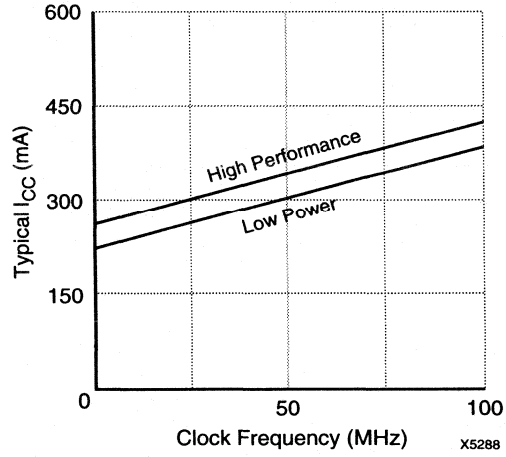


Figure 2. Typical I_{CC} vs Frequency for XC73108

Notice: The information contained in this data sheet pertains to products in the initial production phases of development. These specifications are subject to change without notice. Verify with your local Xilinx sales office that you have the latest data sheet before finalizing a design.

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage with respect to GND	-0.5 to 7.0	V
V_{IN}	DC Input voltage with respect to GND	-0.5 to 7.0	V
V_{TS}	Voltage applied to 3-state output with respect to GND	-0.5 to 7.0	V
T_{STG}	Storage temperature	-65 to +150	°C
T_{SOL}	Maximum soldering temperature (10s @ 1/16 in. = 1.5 mm)	+260	°C

Warning: *Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Recommended Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.*

Recommended Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CCINT}/ V_{CCIO}	Supply voltage relative to GND @ 5 V Commercial T_A = 0° C to 70° C	4.75	5.25	V
	Supply voltage relative to GND @ 5 V Industrial T_A = -40° C to 85° C	4.5	5.5	V
	Supply voltage relative to GND @ 5 V Military T_C = -55° C to 125° C	4.5	5.5	V
V_{CCIO}	I/O supply voltage relative to GND @ 3.3 V	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{IH}	High-level input voltage	2.0	V_{CC}+0.3	V
V_O	Output voltage	0	V_{CCIO}	V

DC Characteristics Over Recommended Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL High-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V High-level output voltage	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V Low-level output voltage	I_{OL} = 24 mA (FO) I_{OL} = 12 mA (I/O) V_{CC} = Min		0.5	V
	3.3 V Low-level output voltage	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output high-Z leakage current	V_{CC} = Max V_O = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance for Input and I/O pins	V_{IN} = GND f = 1.0 MHz		10.0	pF
C_{IN}	Input capacitance for global control pins (FCLK0, FCLK1, FCLK2, FOE0, FOE1)	V_{IN} = GND f = 1.0 MHz		15.0	pF
C_{OUT}(1)	Output capacitance	V_O = GND f = 1.0 MHz		20.0	pF
I_{CC1}(2)	Supply Current (low power mode)	V_{IN} = V_{CC} or GND V_{CCINT} = V_{CCIO} = 5 V f = 1.0 MHz @ 25°C	227 mA Typ		

Notes: 1. Sample tested
2. Measured with device programmed as six 16-bit counters

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{RESET}	Configuration completion time (to outputs operational) following assertion of Master Reset			300	µs

Fast Function Block (FFB) External AC Characteristics

Symbol	Parameter	XC73108-10 (Com Only)		XC73108-12 (Com/Ind Only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{CF}	Max count frequency (1, 2)	100.0		80.0		66.7		50.0		MHz
t_{SUF}	Fast input setup time before FCLK ↑ (1)	5.0		6.0		7.0		10.0		ns
t_{HF}	Fast input hold time after FCLK ↑	0		0		0		0		ns
t_{COF}	FCLK ↑ to output valid		8.0		9.0		12.0		15.0	ns
t_{PDFO}	Fast input to output valid (1, 2)		10.0		12.0		15.0		20.0	ns
t_{PDFU}	I/O to output valid (1, 2)		19.0		22.0		27.0		35.0	ns
t_{CWF}	Fast clock pulse width	5.0		5.5		6.0		6.0		ns

High-Density Function Block (FB) External AC Characteristics

Symbol	Parameter	XC73108-10 (Com Only)		XC73108-12 (Com/Ind Only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_C	Max count frequency (1, 2)	62.5		55.6		45.5		35.7		MHz
t_{SU}	I/O setup time before FCLK ↑ (1, 2)	16.0		18.0		22.0		28.0		ns
t_H	I/O hold time after FCLK ↑	0		0		0		0		ns
t_{CO}	FCLK ↑ to output valid		10.0		12.0		15.0		20.0	ns
t_{PSU}	I/O setup time before p-term clock ↑ (2)	6.0		7.0		9.0		12.0		ns
t_{PH}	I/O hold time after p-term clock ↑	0		0		0		0		ns
t_{PCO}	P-term clock ↑ to output valid		20.0		23.0		28.0		36.0	ns
t_{PD}	I/O to output valid (1, 2)		25.0		30.0		36.0		45.0	ns
t_{CW}	Fast clock pulse width	5.0		5.5		6.0		6.0		ns
t_{PCW}	P-term clock pulse width	6.0		7.5		8.5		12.0		ns

Notes:
1. This parameter is given for the high-performance mode. In low-power mode, this parameter is increased due to additional logic delay of $t_{FLOGILP} - t_{FLOGI}$ or $t_{LOGILP} - t_{LOGI}$.
2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.

Fast Function Block (FFB) Internal AC Characteristics

Symbol	Parameter	XC73108-10 (Com Only)		XC73108-12 (Com/Ind Only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{FLOGI}	FFB logic array delay (2)		1.5		2.0		2.0		3.0	ns
$t_{FLOGILP}$	Low-power FFB logic array delay (2)		5.5		7.0		8.0		11.0	ns
t_{FSUI}	FFB register setup time	2.5		3.0		4.0		6.0		ns
t_{FHI}	FFB register hold time	2.5		3.0		3.0		4.0		ns
t_{FCOI}	FFB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{FPDI}	FFB register pass through delay		0.5		1.0		1.0		2.0	ns
t_{FAOI}	FFB register async. set delay		2.5		3.0		4.0		6.0	ns
t_{PTXI}	FFB p-term assignment delay		1.0		1.2		1.5		2.0	ns
t_{FFD}	FFB feedback delay		5.0		6.5		8.0		10.0	ns

High-Density Function Block (FB) Internal AC Characteristics

Symbol	Parameter	XC73108-10 (Com Only)		XC73108-12 (Com/Ind Only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{LOGI}	FB logic array delay (2)		3.5		4.0		5.0		6.0	ns
t_{LOGILP}	Low power FB logic delay (2)		7.5		9.0		11.0		14.0	ns
t_{SUI}	FB register setup time	2.5		3.0		4.0		6.0		ns
t_{HI}	FB register hold time	3.5		4.0		5.0		6.0		ns
t_{COI}	FB register clock-to-output delay		1.0		1.0		1.0		1.0	ns
t_{PDI}	FB register pass through delay		2.5		4.0		4.0		4.0	ns
t_{AOI}	FB register async. set/reset delay		3.0		4.0		5.0		7.0	ns
t_{RA}	Set/reset recovery time before FCLK ↑	19.0		21.0		25.0		31.0		ns
t_{HA}	Set/reset hold time after FCLK ↑	0		0		0		0		ns
t_{PRA}	Set/reset recovery time before p-term clock ↑	10.0		12.0		15.0		20.0		ns
t_{PHA}	Set/reset hold time after p-term clock ↑	6.0		8.0		9.0		12.0		ns
t_{PCI}	FB p-term clock delay		0		0		0		0	ns
t_{OEI}	FB p-term output enable delay		4.0		5.0		7.0		9.0	ns
t_{CARY8}	ALU carry delay within 1 FB (3)		6.0		8.0		12.0		15.0	ns
t_{CARYFB}	Carry lookahead delay per additional Functional Block (3)		1.5		2.0		3.0		4.0	ns

Notes: 2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.

3. Arithmetic carry delays are measured as the increase in required set-up time to adjacent Macrocell(s) for adder with registered outputs.

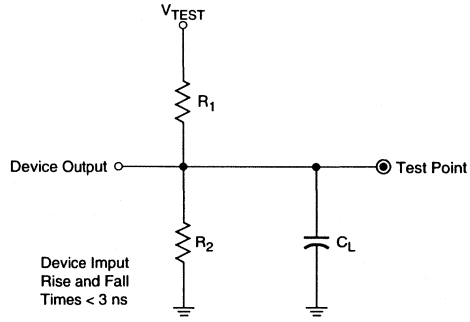
I/O Block External AC Characteristics

Symbol	Parameter	XC73108-10 (Com Only)		XC73108-12 (Com/Ind Only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
f_{IN}	Max pipeline frequency (input register to FFB or FB register) (2)	62.5		55.6		45.5		35.7		MHz
t_{SUIN}	Input register/latch setup time before FCLK ↑	5.0		6.0		7.0		10.0		ns
t_{HIN}	Input register/latch hold time after FCLK ↑	0		0		0		0		ns
t_{COIN}	FCLK ↑ to input register/latch output		3.5		4.0		5.0		6.0	ns
t_{CESUIN}	Clock enable setup time before FCLK ↑	7.0		8.0		10.0		12.0		ns
t_{CEHIN}	Clock enable hold time after FCLK ↑	0		0		0		0		ns
t_{CWHIN}	FCLK pulse width high time	5.0		5.5		6.0		6.0		ns
t_{CWLIN}	FCLK pulse width low time	5.0		5.5		6.0		6.0		ns

Internal AC Characteristics

Symbol	Parameter	XC73108-10 (Com Only)		XC73108-12 (Com/Ind Only)		XC73108-15		XC73108-20		Units
		Min	Max	Min	Max	Min	Max	Min	Max	
t_{IN}	Input pad and buffer delay		3.5		4.0		5.0		6.0	
t_{FOUT}	FFB output buffer and pad delay		4.5		5.0		7.0		9.0	
t_{OUT}	FB output buffer and pad delay		6.5		8.0		10.0		14.0	
t_{UIM}	Universal Interconnect Matrix delay		9.0		10.0		12.0		15.0	
t_{FOEI}	Fast output enable/disable buffer delay		10.0		12.0		15.0		20.0	
t_{FCLKI}	Fast clock buffer delay		2.5		3.0		4.0		5.0	

Note: 2. Specifications account for logic paths that use the maximum number of available product terms for a given Macrocell.



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
FO	5.0 V	5.0 V	160 Ω	120 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF
O	5.0 V	5.0 V	310 Ω	195 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X3491

Figure 3. AC Load Circuit

XC73108 Pinouts

PQ160	PG144 PB225	PC84	Pin Description
1	D3	–	V_{CCIO}
2	C2	75	O/CKEN1
3	–	–	N/C
4	B1	–	FO
5	–	–	N/C
6	D2	76	O/F_{OE0}
7	E3	–	O
8	C1	77	O/F_{OE1}
9	E2	–	O
10	D1	78	V_{CCINT}/V_{PP}
11	F3	79	I/FI
12	F2	–	I/O/FI
13	E1	80	I/FI
14	G2	–	I/O/FI
15	G3	81	I/FI
16	F1	–	I/O/FI
17	G1	82	I/FI
18	H2	83	I/FI
19	H1	84	I/FI
20	H3	–	GND
21	J3	1	MRX
22	J1	2	I/FI
23	K1	3	I/FI
24	J2	4	I/FI
25	K2	–	O
26	K3	5	I/FI
27	L1	-	O
28	L2	6	I/FI
29	M1	–	I/O/FI
30	N1	7	I/FI
31	M2	8	GND
32	L3	–	I/O/FI
33	N2	9	O/F_{CLK0}
34	P1	–	O
35	M3	10	O/F_{CLK1}
36	N3	–	FO
37	P2	11	I/O/FI
38	–	–	N/C
39	–	–	N/C
40	R1	–	GND

PQ160	PG144 PB225	PC84	Pin Description
41	N4	–	V_{CCIO}
42	P3	12	O/F_{CLK2}
43	R2	–	I/O
44	P4	13	FO
45	N5	–	I/O
46	R3	–	V_{CCINT}
47	P5	14	FO
48	R4	–	I/O
49	N6	15	FO
50	P6	–	I/O
51	R5	16	GND
52	–	–	N/C
53	–	–	N/C
54	P7	17	FO
55	N7	–	I/O
56	R6	18	FO
57	R7	–	I/O/FI
58	P8	19	FO
59	R8	20	FO
60	N8	21	FO
61	N9	22	V_{CCIO}
62	R9	23	I/O
63	R10	24	I/O
64	P9	25	I/O
65	–	–	N/C
66	–	–	N/C
67	P10	–	I/O
68	N10	26	I/O
69	R11	–	I/O/FI
70	P11	27	GND
71	R12	28	I/O/FI
72	R13	–	I/O/FI
73	P12	29	I/O/FI
74	N11	–	I/O
75	P13	30	I/O/FI
76	R14	–	I/O
77	N12	31	I/O
78	N13	–	I/O
79	P14	32	I/O
80	R15	–	GND

Note: With the XC73108 in the 225-pin ball grid array package, only 144 of the solder balls are connected, the remaining solder balls should be left unconnected

XC73108 Pinouts (continued)

PQ160	PG144 PB225	PC84	Pin Description
81	M13	–	V_{CCIO}
82	N14	33	I/O
83	–	–	N/C
84	P15	–	I/O
85	–	–	N/C
86	M14	34	I/O
87	L13	–	I/O
88	N15	35	I/O
89	L14	–	I/O
90	M15	36	I/O
91	K13	–	I/O/FI
92	K14	37	I/O
93	L15	–	I/O/FI
94	J14	38	V_{CCINT}
95	J13	39	I/O
96	K15	–	I/O/FI
97	J15	40	I/O/FI
98	H14	41	I/O/FI
99	H15	–	GND
100	H13	42	GND
101	G13	43	I/O/FI
102	G15	44	I/O
103	F15	45	I/O
104	G14	46	I/O
105	F14	–	I/O
106	F13	47	I/O
107	E15	–	I/O
108	E14	48	I/O
109	D15	–	I/O
110	C15	49	GND
111	D14	50	I/O
112	E13	–	I/O
113	C14	51	I/O
114	B15	–	I/O
115	D13	52	I/O
116	C13	–	I/O/FI
117	B14	53	I/O/FI
118	–	–	N/C
119	–	–	N/C
120	A15	–	GND

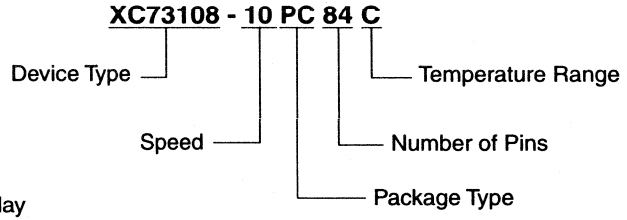
PQ160	PG144 PB225	PC84	Pin Description
121	C12	–	V_{CCIO}
122	B13	54	I/O/FI
123	A14	–	I/O
124	B12	55	I/O/FI
125	C11	–	I/O/FI
126	A13	56	I/O
127	B11	–	GND
128	A12	–	I/O/FI
129	C10	–	I/O/FI
130	B10	–	I/O
131	–	–	N/C
132	–	–	N/C
133	A11	–	I/O/FI
134	B9	57	I/O
135	C9	58	I/O
136	A10	59	I/O
137	A9	60	GND
138	B8	61	I/O/FI
139	A8	62	I/O/FI
140	C8	63	I/O/FI
141	C7	64	V_{CCIO}
142	A7	65	FO
143	A6	66	FO
144	B7	67	FO
145	B6	–	I/O/FI
146	C6	68	FO
147	A5	–	I/O
148	B5	69	FO
149	–	–	N/C
150	–	–	N/C
151	A4	–	I/O
152	A3	70	FO
153	B4	–	I/O
154	C5	71	FO
155	B3	–	I/O
156	A2	72	FO
157	C4	73	V_{CCINT}
158	C3	–	I/O
159	B2	74	O/$\overline{CKEN0}$
160	A1	–	160

For a detailed description of the device architecture, see the XC7300 CMOS EPLD Family data sheet, page 3-5 through 3-11.

For a detailed description of the device timing, see pages 3-13, 3-14 and 3-48 through 3-50.

For package physical dimensions and thermal data, see Section 4.

Ordering Information



Speed Options

-20	20 ns pin-to-pin delay
-15	15 ns pin-to-pin delay
-12	12 ns pin-to-pin delay (commercial and industrial only)
-10	10 ns pin-to-pin delay (commercial only)

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	-40°C to 85°C
M	Military	-55°C to 125°C (Case)

Packaging Options

PC84	84-Pin Plastic Leaded Chip Carrier
WC84	84-Pin Windowed Ceramic Leaded Chip Carrier
PG144	144-Pin Windowed Pin-Grid-Array
PQ160	160-Pin Plastic Quad Flat Pack
PB225	225-pin Plastic Ball-Grid-Array
WB225	225-pin Windowed Ceramic Ball-Grid-Array

Component Availability

Pins	44		68		84			144	160	184	225	
Type	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Ceramic PGA	Ceramic PGA	Plastic PQFP	Ceramic PGA	Plastic BGA	Ceramic BGA
Code	PC44	WC44	PC68	WC68	PC84	WC84	PG84	PG144	PQ160	PG184	PB225	WB225
XC73108 -20					CI	CI(M)		CI(M)	CI		(CI)	(CI)
XC73108 -15					CI	CI(M)		CI(M)	CI		(CI)	(CI)
XC73108 -12					CI	CI		CI	CI		(CI)	(CI)
XC73108 -10					(C)	(C)		(C)	(C)		(C)	(C)

C = Commercial = 0° to +70°C I = Industrial = -40° to 85°C

Parenthesis indicate future product plans

X5279

XC73144
144-Macrocell CMOS EPLD

Advance Product Information

Features

- High-Performance EPLD
  - 10 ns pin-to-pin delay
  - 100 MHz maximum clock frequency
- Advanced Dual-Block architecture
  - Four Fast Function Blocks
  - 12 High-Density Function Blocks
- 100% interconnect matrix
- High-Speed arithmetic carry network
  - 1 ns ripple-carry delay per bit
  - 35 MHz 16-bit accumulators
- 144 Macrocells with programmable I/O architecture
- Up to 90 inputs programmable as direct, latched, or registered
- 36 outputs with 24 mA drive
- 3.3 V or 5 V I/O operation
- Meets JEDEC Standard (8-1A) for 3.3 V ±0.3 V
- Power management options
- Multiple security bits for design protection
- 184-pin pin-grid-array package
- 225-pin ball-grid-array package

General Description

The XC73144 is a member of the Xilinx Dual-Block EPLD family. It consists of four Fast Function Blocks and twelve High-Density Function Blocks interconnected by a central Universal Interconnect Matrix (UIM).

The sixteen Function Blocks in the XC73144 are PAL-like structures, complete with programmable product term arrays and programmable multilevel Macrocells. Each Function Block receives 24 inputs, contains nine Macrocells configurable for registered or combinatorial logic and produces nine outputs which feedback to the UIM and output pins.

The Universal Interconnect Matrix connects the Function Blocks to each other and to all input pins, providing 100% connectivity between the Function Blocks. This allows logic functions to be mapped into the Function Blocks and interconnected without routing restrictions.

In addition, the XC73144 includes a programmable power management feature to specify high-performance or low-power operation on an individual Macrocell-by-Macrocell basis. Unused Macrocells are automatically turned off to minimize power dissipation. Designers can operate speed-critical paths at maximum performance, while non-critical paths dissipate less power.

Xilinx development software (XEPLD) supports all members of XC7300 family. The designer can create, implement, and verify digital logic circuits for EPLD devices using the Xilinx XEPLD Development System. Designs can be represented as schematics consisting of XEPLD library components, as behavioral descriptions, or as a mixture of both. The XEPLD translator automatically performs logic optimization, collapsing, mapping and routing without user intervention. After compiling the design, XEPLD translator produces documentation for design analysis and creates a programming file to configure the device.

The following lists some of the XEPLD Development System features.

- Familiar design approach similar to TTL and PLD techniques
- Converts netlist to fuse map in minutes using a 386/486 PC or workstation platform
- Interfaces to standard third-party CAE schematics, simulation tools, and behavioral languages
- Timing simulation using Viewsim, OrCAD VST, Mentor, LMC and other tools compatible with the Xilinx Netlist Format (XNF)

XC7200A EPLD Family

Table of Contents

Overview

The Xilinx XC7200A family offers three advantages.

- Each device contains dedicated high-speed arithmetic carry logic for efficient implementation of fast adders, subtractors, accumulators, and magnitude comparators.
- Unrestricted Universal Interconnect Matrix (UIM) for guaranteed interconnect of all internal logic resources.
- Each programmable input structure can be configured as either direct, latched, or registered in a flip-flop.

	XC7236A	XC7272A
Typical 22V10 Equivalent	4	8
Number of Macrocells	36	72
Number of Function Blocks	4	8
Number of Flip-Flops	68	126
Number of Signal Pins	36	72

Component Availability

Pins	44		68		84		
Type	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Ceramic PGA
Code	PC44	WC44	PC68	WC68	PC84	WC84	PG84
-25	CI	CI(M)					
XC7236A -20	CI	CI(M)					
-16	CI	CI					
-25			CI	CI	CI	CI(M)	CI
XC7272A -20			CI	CI	CI	CI	CI
-16			CI	CI	CI	CI	CI

C = Commercial = 0° to +70°C I = Industrial = -40° to 85°C

Parenthesis indicate future product plans

X3480

XC7236A
36-Macrocell CMOS EPLD

Preliminary Product Specifications

Features

- Second-Generation High Density Programmable Logic Device
- UV-erasable CMOS EPROM technology
- 36 Macrocells, grouped into four Function Blocks, interconnected by a programmable Universal Interconnect Matrix (UIM)
- Each Function Block contains a programmable AND-array with up to 24 complementary inputs, providing up to 17 product terms per Macrocell
- Enhanced logic features
  - Arithmetic Logic Unit in each Macrocell
  - Dedicated fast carry network between Macrocells
  - Wide AND capability in the Universal Interconnect Matrix
- Identical timing for all interconnect paths and for all Macrocell logic paths
- 36 signal pins
  - 30 I/Os, 2 inputs, 4 outputs
- Each input is programmable
  - Direct, latched, or registered
- I/O operation at 3.3 V or 5 V
- Meets JEDEC Standard (8-1A) for 3.3 V ± 0.3 V
- Three high-speed, low-skew global clock inputs
- 44-pin plastic and windowed ceramic leaded chip carrier packages

General Description

The XC7236A is a second-generation High Density Programmable Logic Device that combines the classical features of the PAL-like EPLD architecture with innovative systems-oriented logic enhancements. This favors the implementation of fast state machines, large synchronous counters and fast arithmetic, as well as multi-level general-purpose logic. Performance, measured in achievable system clock rate and critical delays, is not only predictable, but independent of physical logic mapping, interconnect routing, and resource utilization. Performance, therefore, remains invariant between design iterations. The propagation delay through interconnect and logic is constant for any function implemented in any one of the output Macrocells.

The functional versatility of the traditional programmable logic array architecture is enhanced through additional gating and control functions available in an Arithmetic Logic Unit (ALU) in each Macrocell. Dedicated fast arithmetic carry lines running directly between adjacent Macrocells and Function Blocks support fast adders, subtractors and comparators of any length up to 36 bits.

This additional ALU in each Macrocell can generate any combinatorial function of two sums of products, and it can generate and propagate arithmetic-carry signals between adjacent Macrocells and Functional Blocks.

The Universal Interconnect Matrix (UIM) facilitates unrestricted, fixed-delay interconnects from all device inputs and Macrocell outputs to any Function Block AND-array input. The UIM can also perform a logical AND across any number of its incoming signals on the way to any Function Block, adding another level of logic without additional delay. This supports bidirectional loadable synchronous counters of any size up to 36 bits, operating at the specified maximum device frequency

As a result of these logic enhancements, the XC7236A can deliver high performance even in designs that combine large numbers of product terms per output, or need more layers of logic than AND-OR, or need a wide AND function in some of the product terms, or perform wide arithmetic functions.

Automated design mapping and optimization is supported by Xilinx XEPLD development software based on design capture using third-party schematic entry tools, PLD compilers or direct text-based equation files. Design mapping is completed in a few minutes on a PC or Worstation.

Architectural Overview

Figure 1 shows the XC7236A structure. Four Function Blocks (FBs) are all interconnected by a central UIM. Each FB receives 21 signals from the UIM and each FB produces nine signals back into the UIM. All device inputs are also routed via the UIM to all FBs. Each FB contains nine output Macrocells (MCs) that draw from a programmable AND array driven by the 21 signals from the UIM. Most Macrocells drive a 3-state chip output; all feed back into the UIM.

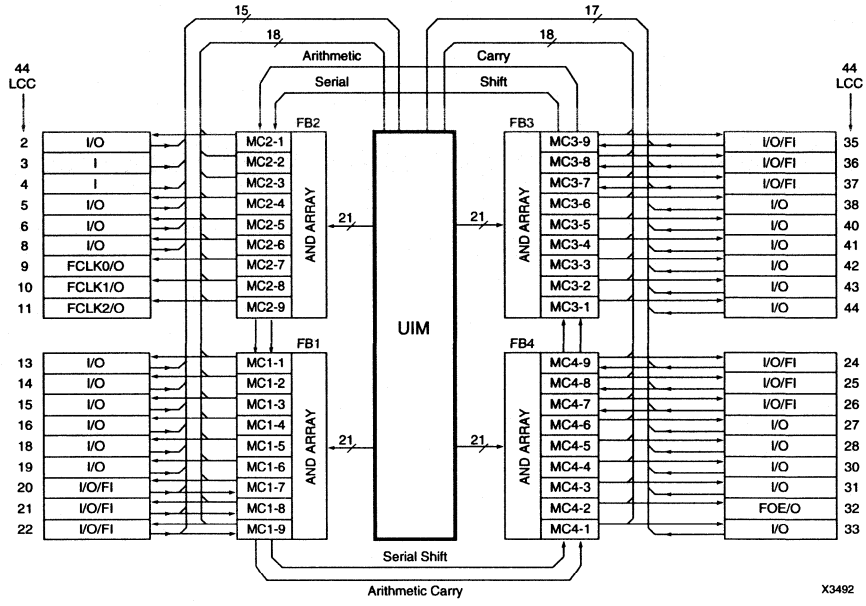


Figure 1. XC7236A Architecture

Function Blocks and Macrocells

The XC7236A contains 36 Macrocells with identical structure, grouped into four Function Blocks of nine Macrocells each. Each Macrocell is driven by product terms derived from a programmable AND array in the Function Block. The AND array in each Function Block receives 21 signals and their complements from the UIM. In three Function Blocks, the AND array receives three additional inputs and their complements directly from FastInput (FI) pins, thus offering faster logic paths.

Five product terms are private to each Macrocell; an additional 12 product terms are shared among the nine Macrocells in each Function Block. Four of the private product terms can be selectively ORed together with up to four of the shared product terms, and drive the D1 input to the ALU. The other input, D2, to the ALU is driven by the OR of the fifth private product term and up to eight of the remaining shared product terms.

As a programmable option, four of the private product terms can be used for other purposes. One of the private product terms can be used as a dedicated clock for the flip-flop in the Macrocell. (See the subsequent description of other clocking options.) Another one of the private product terms can be the asynchronous active-High Reset of the Macrocell flip-flop, another one can be the asynchronous active-High Set of the Macrocell flip-flop, and another one can be the Output Enable signal.

As a configuration option, the Macrocell output can be fed back and ORed into the D2 input to the ALU after being ANDed with three of the shared product terms to implement counters and toggle flip-flops.

The ALU has two programmable modes: In the *logic mode,* it is a 2-input function generator, a 4-bit look-up table, that can be programmed to generate any Boolean function of its two inputs. It can OR them, widening the OR function to max 17 inputs; it can AND them, which means that one sum of products can be used to mask the other; it can XOR them, toggling the flip-flop or comparing the two sums of products. Either or both of the sum-of-product inputs to the ALU can be inverted and either or both can be ignored. The ALU can implement one additional layer of logic without any speed penalty.

In the *arithmetic mode,* the ALU block in each Macrocell can be programmed to generate the arithmetic sum or difference of two operands, combined with a carry signal coming from the next lower Macrocell. It also feeds a carry output to the next higher Macrocell. This carry propagation chain crosses the boundaries between Function Blocks. This dedicated carry chain overcomes the inherent speed and density problems of the traditional EPLD architecture, when trying to perform arithmetic functions.

The ALU output drives the D input of the Macrocell flip-flop. Each flip-flop has several programmable options. One option is to eliminate the flip-flop by making it transparent, which makes the Q output identical with the D input, independent of the clock. Otherwise, the flip-flop operates in the conventional manner, triggered by the rising edge on its clock input.

The clock source is programmable and is either the dedicated product term mentioned earlier, or one of two global FastCLK signals (FLCK0 or FLCK1) that are distributed with short delay and minimal skew over the whole chip.

The asynchronous Set and Reset (Clear) inputs override the clocked operation. If both asynchronous inputs are active simultaneously, Reset overrides Set. Upon power-up, each Macrocell flip-flop can be preloaded with either 0 or 1.

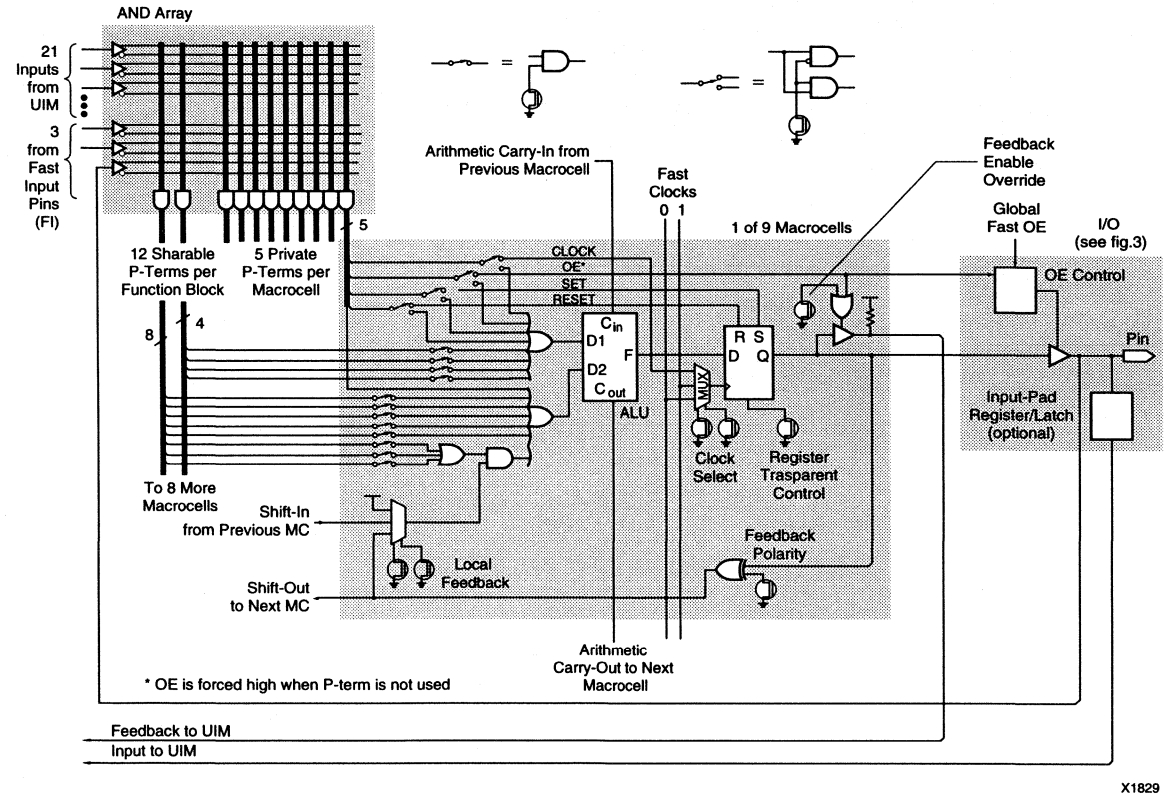


Figure 2. Function Block and Macrocell Schematic

In addition to driving a chip output pin, the Macrocell output is also routed back as an input to the UIM. One private product term can be configured to control the Output Enable of the output pin driver and/or the feedback to the UIM. If configured to control UIM feedback, then when the OE product-term is de-asserted, the UIM feedback line is forced High and thus disabled.

Universal Interconnect Matrix

The UIM receives 68 inputs: 36 from the Macrocell feedbacks, 30 from bidirectional I/O pins, and 2 from dedicated input pins. Acting as an unrestricted crossbar switch, the UIM generates 84 output signals, 21 to each Function Block.

Any one of the 68 inputs can be programmed to be connected to any number of the 84 outputs. The delay through the array is constant, independent of the apparent routing distance, the fan-out, fan-in, or routing complexity.

Routability is not an issue in that any UIM input can drive any UIM output or multiple outputs without additional delay.

When multiple inputs are programmed to be connected to the same output, this output becomes the AND of the input signals if the levels are interpreted as active High. By choosing the appropriate signal inversion at the input pin, Macrocell outputs and Function Block AND-array input, this AND-logic can also be used to implement a NAND, OR, or NOR function. This offers an additional level of logic without any speed penalty.

A Macrocell feedback signal that is disabled by the output enable product term represents a High input to the UIM. Several such Macrocell outputs programmed onto the same UIM output thus emulating a 3-state bus line. If one of the Macrocell outputs is enabled, the UIM output assumes that same level.

Outputs

Thirty-four of the 36 Macrocell drive chip outputs directly through individually programmable inverters followed by 3-state output buffers; each can be individually controlled by the Output Enable product term mentioned above. An additional configuration option disables the output permanently. One dedicated FastOE input can also be configured to control any of the chip outputs instead of, or in conjunction with the individual OE product term.

Inputs

Each signal input to the chip is programmable as either direct, latched, or registered in a flip flop. Latch and flip-flop can be programmed with either of two FastCLK signals as latch enable or clock. The two FastCLK signals are FCLK0 and a global choice of either FCLK1 or FCLK2. Latches are transparent when FastCLK is High, and flip-flops clock on the rising edge of FastCLK. Registered inputs allow high system clock rates by pipelining the inputs before they

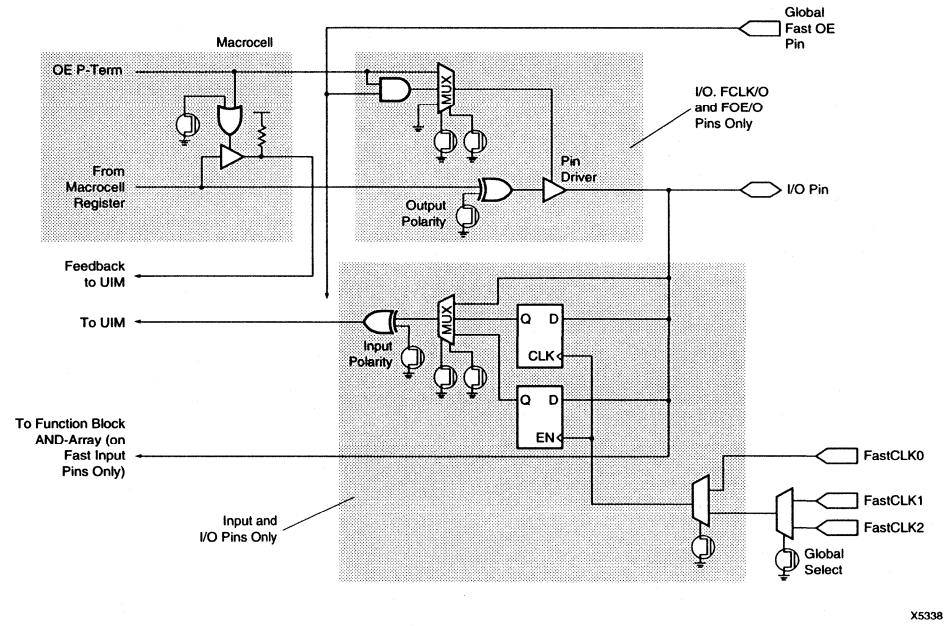


Figure 3. Input/Output Schematic

incur the combinatorial delay in the device, provided the one-clock-period pipeline latency is acceptable.

The direct, latched, or registered inputs then drive the UIM. There is no propagation-delay difference between pure inputs and I/O inputs.

3.3 V or 5 V Interface configuration

The XC7236A can be used in systems with two different supply voltages, 5 V or 3.3 V. The device has separate V_{CC} connections to the internal logic and input buffers (V_{CCINT}) and to the I/O output drivers (V_{CCIO}). V_{CCINT} is always connected to a nominal +5 V supply, but V_{CCIO} may be connected to either +5 V or +3.3 V, depending on the output interface requirement.

When V_{CCIO} is connected to +5 V, the input thresholds are TTL levels, and thus compatible with 5 V or 3.3 V logic, and the output high levels are compatible with 5 V systems. When V_{CCIO} is connected to 3.3 V, the input thresholds are still TTL levels, and the outputs pull up to the 3.3 V rail. This makes the XC7236A ideal for interfacing directly to 3.3 V components. In addition, the output structure is designed such that the I/O can also safely interface to a mixed 3.3-V or 5-V bus.

Programming and Using the XC7236A

The features and capabilities described above are used by the Xilinx development software to program the device according to the specification given either through schematic entry, or through a behavioral description expressed in Boolean equations.

The user can specify a security bit that prevents any reading of the programming bit map after the device has been programmed and verified.

The device is programmed in a manner similar to an EPROM (ultra-violet light erasable read-only memory) using the Intel Hex format. Programming support is available from a number of programmer manufacturers. The UIM connections and Function Block AND-array connections are made directly by non-volatile EPROM cells. Other control bits are read out of the EPROM array and stored into latches just after power-up. This method, common among EPLD devices, requires application of a master-reset signal delayed at least until V_{CC} has reached the required operating voltage. This can be achieved using a simple capacitor and pull-up resistor on the MR pin (the RC product should be larger than twice the V_{CC} rise time). The power-up or reset signal initiates a self-timed configuration period lasting about 350 µs (t_{RESET}), during which all device outputs remain disabled and programmed preload state values are loaded into the Macrocell registers.

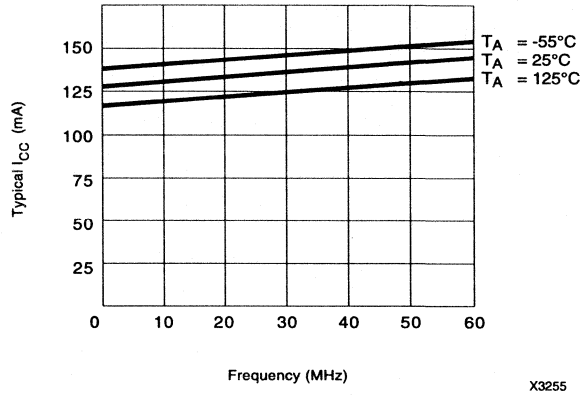


Figure 4. Typical I_{CC} vs Frequency for XC7236A Configured as Eight 4-bit Counters
(V_{CC} = +5.0 V, V_{IN} = 0 or 5 V, all outputs open)

Unused input and I/O pins should be tied to ground or Vcc or some valid logic level. This is common practice for all CMOS devices to avoid dissipating excess current through the input pad circuitry.

The recommended decoupling capacitance on the three V_{CC} pins should total 1 µF using high-speed (tantalum or ceramic) capacitors.

Absolute Maximum Ratings

Symbol	Parameter	Value	Units
V_{CC}	Supply voltage relative to GND	–0.5 to 7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to 7.0	V
V_{TS}	Voltage applied to 3-state output	–0.5 to 7.0	V
T_{STG}	Storage temperature	–65 to + 150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in. = 1.5 mm)	+ 260	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

Symbol	Parameter	Min	Max	Units
V_{CCINT}	Supply voltage relative to GND Commercial T_A = 0°C to 70°C	4.75	5.25	V
V_{CCIO}	Supply voltage relative to GND Industrial T_A = –40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND Military T_C= –55°C to 125°C	4.5	5.5	V
V_{CCIO}	I/O supply voltage 3.3 V	3.0	3.6	V
V_{IL}	Low-level input voltage	0	0.8	V
V_{IH}	High-level input voltage	2.0	V_{CC}+ 0.3	V
V_O	Output voltage	0	V_{CCIO}	V

DC Characteristics Over Operating Conditions

Symbol	Parameter	Test Conditions	Min	Max	Units
V_{OH}	5 V TTL high-level output voltage	I_{OH} = -4.0 mA V_{CC} = Min	2.4		V
	3.3 V high-level output	I_{OH} = -3.2 mA V_{CC} = Min	2.4		V
V_{OL}	5 V low-level output voltage	I_{OL} = 12 mA V_{CC} = Min		0.5	V
	3.3 V low-level output voltage	I_{OL} = 10 mA V_{CC} = Min		0.4	V
I_{CC}	Supply Current	V_{IN}= 0 V V_{CC} = Max f = 0 MHz	126 mA Typ		
I_{IL}	Input leakage current	V_{CC} = Max V_{IN} = GND or V_{CCIO}		±10.0	µA
I_{OZ}	Output High-Z leakage current	V_{CC} = Max V_O = GND or V_{CCIO}		±10.0	µA
C_{IN}	Input capacitance (sample tested)	V_{IN} = GND f = 1.0 MHz		10.0	pF

AC Timing Requirements

Symbol	Parameter	Fig.	XC7236A-25		XC7236A-20		*XC7236A-16		Units
			Min	Max	Min	Max	Min	Max	
f_{CYC} (Note 1)	Max sequential toggle frequency (with feedback) using FastCLK	6	40		50		60		MHz
f_{CYC1} (Note 1)	Max sequential toggle frequency (with feedback) using a Product-Term clock	6	40		50		60		MHz
f_{CYC4} (Note 5)	Max Macrocell toggle frequency using local feedback and FastCLK		50		50		60		MHz
f_{CLK} (Note 5)	Max Macrocell register transmission frequency (without feedback) using FastCLK		45		50		60		MHz
f_{CLK1} (Note 5)	Max Macrocell register transmission frequency (without feedback) using a Product-Term clock		42		50		60		MHz
f_{CLK2} (Note 5)	Max input register transmission frequency (without feedback) using FastCLK		50		50		60		MHz
f_{CLK3} (Note 1)	Max input register to Macrocell register pipline frequency using FastCLK	7	33		40		60		MHz
t_W	FastCLK pulse width (High/Low)	11	10		8		6		ns
t_{W1}	Product-Term clock width (active/inactive)	11	12		9		7		ns
t_{SU}	Input to Macrocell register set-up time before FastCLK	9	29		24		18		ns
t_H	Input to Macrocell register hold time after FastCLK	9	-7		-4		-4		ns
t_{SU1} (Note 1)	Input to Macrocell register set-up time before Product-Term clock	8	16		14		10		ns
t_{H1}	Input to Macrocell register hold time after Product-Term clock	8	0		0		0		ns
t_{SU2}	Input register/latch set-up time before FastCLK	10	8		8		6		ns
t_{H2}	Input register/latch hold time after FastCLK	10	0		0		0		ns
t_{SU5}	FastInput to Macrocell register set-up time before FastCLK		20		18		15		ns
t_{H5}	FastInput to Macrocell register hold time after FastCLK		0		0		0		ns
t_{WA}	Set/Reset pulse width (active)	11	12		12		10		ns
t_{RA}	Set/Reset input recovery set-up time before FastCLK	11	30		25		20		ns
t_{HA}	Set/Reset input hold time after FastCLK	11	-5		0		0		ns
t_{RA1}	Set/Reset input recovery time before Product-Term clock	11	15		15		12		ns
t_{HA1}	Set/Reset input hold time after P-Term clock	11	9		9		8		ns
t_{HRS}	Product-Term clock width (active/inactive)		10		10		8		ns

*Commercial/Industrial Only

X5208

Propagation Delays

Symbol	Parameter	Fig.	XC7236A-25		XC7236A-20		*XC7236A-16		Units
			Min	Max	Min	Max	Min	Max	
t_{CO}	FastCLK input to register output delay	11	5	14	3	13	3	10	ns
t_{CO1}	P-Term clock input to registered output delay	11	10	30	5	24	5	20	ns
t_{AO}	Set/Reset input to registered output delay	11	10	40	5	32	5	25	ns
t_{PD} (Note 1)	Input to nonregistered output delay	11	10	40	5	32	5	25	ns
t_{OE}	Input to output enable	11	10	32	5	25	5	20	ns
t_{OD}	Input to output disable	11	10	32	5	25	5	20	ns
t_{PD5}	FastInput to non-registered Macrocell output delay		10	31	5	25	5	20	ns
t_{OE5}	FastInput to output enable		5	23	3	20	3	15	ns
t_{OD5}	FastInput to output disable		5	23	3	20	3	15	ns
t_{FOE}	FOE input to output enable		5	15	3	14	3	12	ns
t_{FOD}	FOE input to output disable		5	15	3	14	3	12	ns

*Commercial/Industrial Only

X520

Incremental Parameters

Symbol	Parameter	Fig.	XC7236A-25		XC7236A-20		*XC7236A-16		Units
			Min	Max	Min	Max	Min	Max	
t_{PDT1} (Note 2)	Arithmetic carry delay between adjacent Macrocells	12		1.2		1.2		1	ns
t_{PDT8} (Note 2)	Arithmetic carry delay through 9 adjacent Macrocells in a Function Block	12		6		5		3	ns
t_{PDT9} (Note 2)	Arithmetic carry delay through 10 Macrocells from Macrocell #n to Macrocell #n in next F Block	12		9		6		4	ns
t_{COF1}	Incremental delay from UIM-input (for P-Term clock) to registered Macrocell feedback	13		12		7		5	ns
t_{COF2} (Note 3)	Incremental delay from FastCLK net to latched/registered UIM-input	13		1		1		1	ns
t_{PDF} (Note 1)	Incremental delay from UIM-input to nonregistered Macrocell feedback	13		22		14		10	ns
t_{AOF}	Incremental delay from UIM-input (Set/Reset) to registered Macrocell feedback	13		22		14		10	ns
t_{OEF}, t_{ODF}	Incremental delay from UIM-input (used as output-enable/disable) to Macrocell feedback	13		14		7		5	ns
$t_{IN} + t_{OUT}$ (Note 4)	Propagation delay through unregistered input pad (to UIM) plus output pad driver (from Macrocell)	13		18		18		15	ns

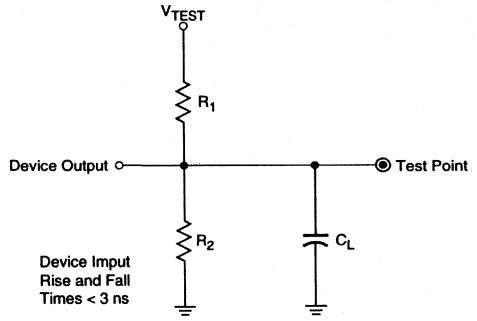
*Commercial/Industrial Only

X5210

Power-up/Reset Timing Parameters

Symbol	Parameter	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{rVCC} (Note 6)	V_{CC} rise time (if MR not used for power-up)			5	µs
t_{RESET}	Configuration completion time (to outputs operational)		350	1000	µs

Notes: 1. Specifications account for logic paths which use the maximum number of available product terms and the ALU.
2. Arithmetic carry delays are measured as the increase in required set-up time to adjacent Macrocell(s) for an adder with registered outputs.
3. Parameter t_{COF2} is derived as the difference between the clock period for pipelining input-to-Macrocell registers ($1/f_{CLK3}$) and the non-registered input set-up time (t_{SU}).
4. Parameter t_{IN} represents the delay from an input or I/O pin to a UIM-input (or from a FastCLK pin to the Fast CLK net); t_{OUT} represents the delay from a Macrocell output (feedback point) to an output or I/O pin. Only the sum of $t_{IN} + t_{OUT}$ can be derived from measurements, e.g., $t_{IN} + t_{OUT} = t_{SU} + t_{CO} - 1/f_{CYC}$.
5. Not tested but derived from appropriate pulse-widths, set-up time and hold-time measurements.
6. Due to the synchronous operation of the power-up reset and the wide range of ways V_{CC} can rise to its steady state, V_{CC} rise *must be* monotonic. Following reset, the Clock, Reset and Set inputs must not be asserted until all applicable input and feedback set-up times are met.



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
O	5.0 V	5.0 V	310 Ω	195 Ω	35 pF
	3.3 V	3.3 V	260 Ω	360 Ω	35 pF

X3489

Figure 5. AC Load Circuit

Timing and Delay Path Specifications

The delay path consists of three blocks that can be connected in series:

- Input Buffer and associated latch or register
- Logic Resource (UIM, AND-array and Macrocell)
- Three-state Output Buffer

All inputs have the same delay, regardless of fan-out or location. All logic resources have the same delay, regardless of logic complexity, interconnect topology or location on the chip. All outputs have the same delay. The achievable clock rate is, therefore, determined only by the input method (direct, latched or registered) and the number of times a signal passes through the combinatorial logic.

Timing and Delay Path Descriptions

Figure 6 defines the max clock frequency (with feedback). Any Macrocell output can be fed back to the UIM as an input for the next clock cycle. The parameters f_{CYC} and f_{CYC1} specify the maximum operating frequency for FastCLK and product-term clock operation respectively.

Figure 7 specifies the max operating frequency (f_{CLK3}) for pipelined operation between the input registers and the Macrocell registers, using FastCLK.

Figure 8 defines the set-up and hold times from the data inputs to the product-term clock used by the output register.

Figure 9 defines the set-up and hold times from the data inputs to the FastCLK used by the output register.

Figure 10 defines the set-up and hold times from the data input to the FastCLK used in an input register.

Figure 11 shows the waveforms for the Macrocell and control paths.

Figure 12 defines the carry propagation delays between Macrocells and between Function Blocks. The parameters describe the delay from the CIN, D1 and D2 inputs of a Macrocell ALU to the CIN input of the adjacent Macrocell ALU. These delays must be added to the standard Macrocell delay path (t_{PD} or t_{SU})to determine the performance of an arithmetic function.

Figure 13 defines the incremental parameters for the standard Macrocell logic paths. These incremental parameters are used in conjunction with pin-to-pin parameters when calculating compound logic path timing. Incremental parameters are derived indirectly from other pin-to-pin measurement.

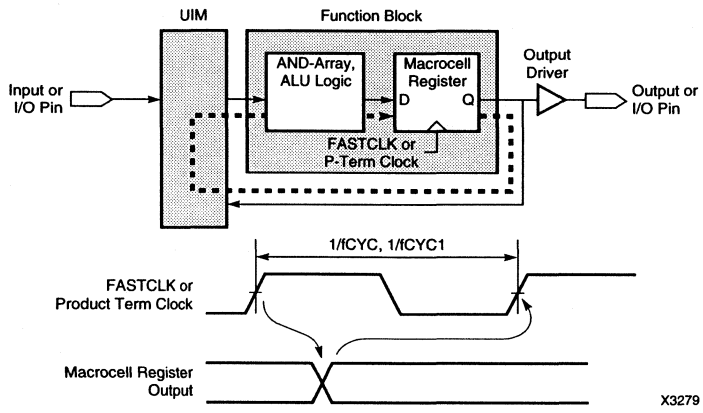


Figure 6. Delay Path Specifications for f_{CYC} and f_{CYC1}

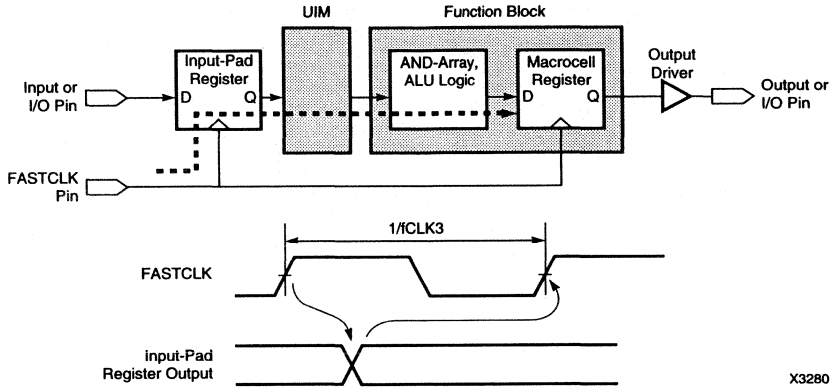


Figure 7. Delay Path Specification for f_{CLK3}

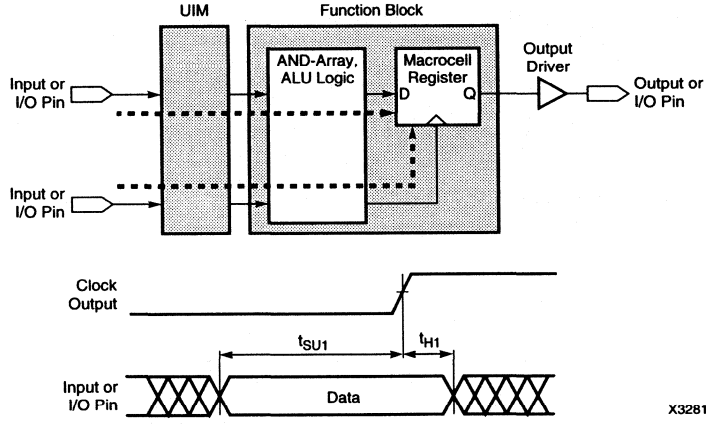


Figure 8. Delay Path Specification for t_{SU1} and t_{H1}

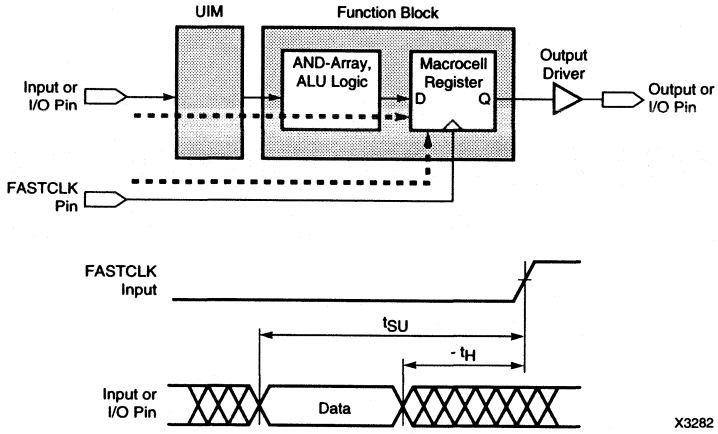


Figure 9. Delay Path Specification for t_{SU} and t_H

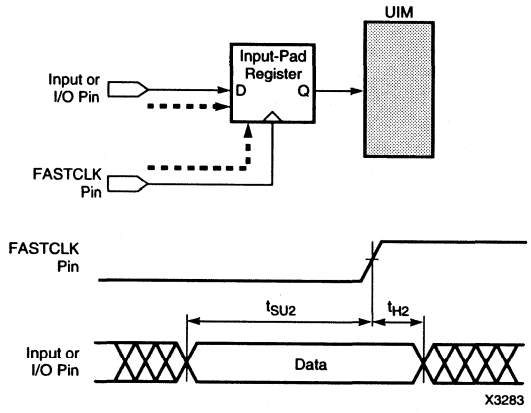


Figure 10. Delay Path Specification for $t_{SU2}$and t_{H2}

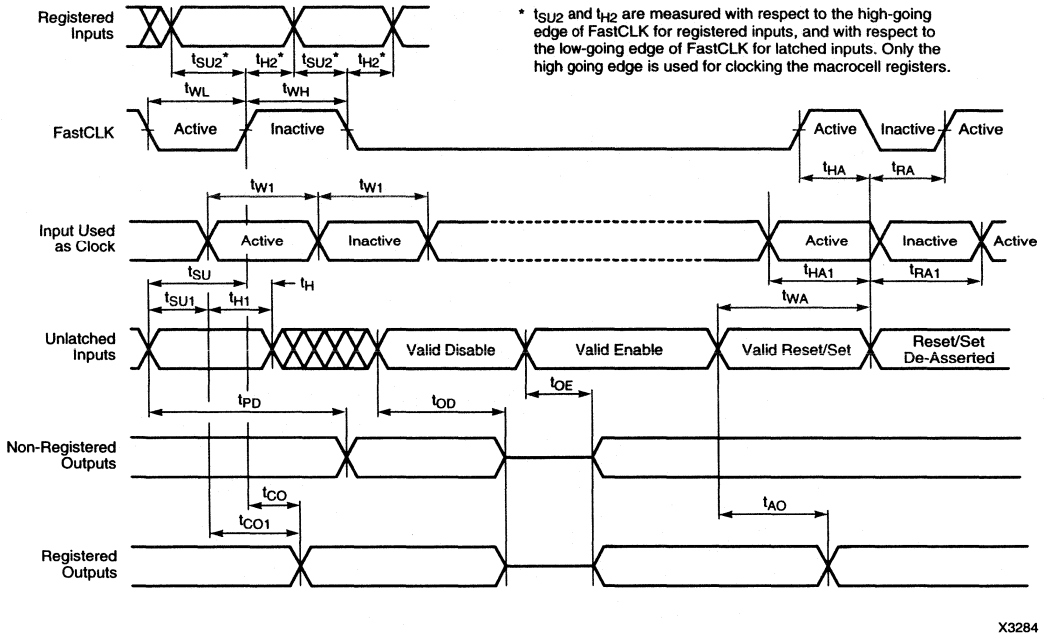


Figure 11. Principal Pin-to-Pin Measurements

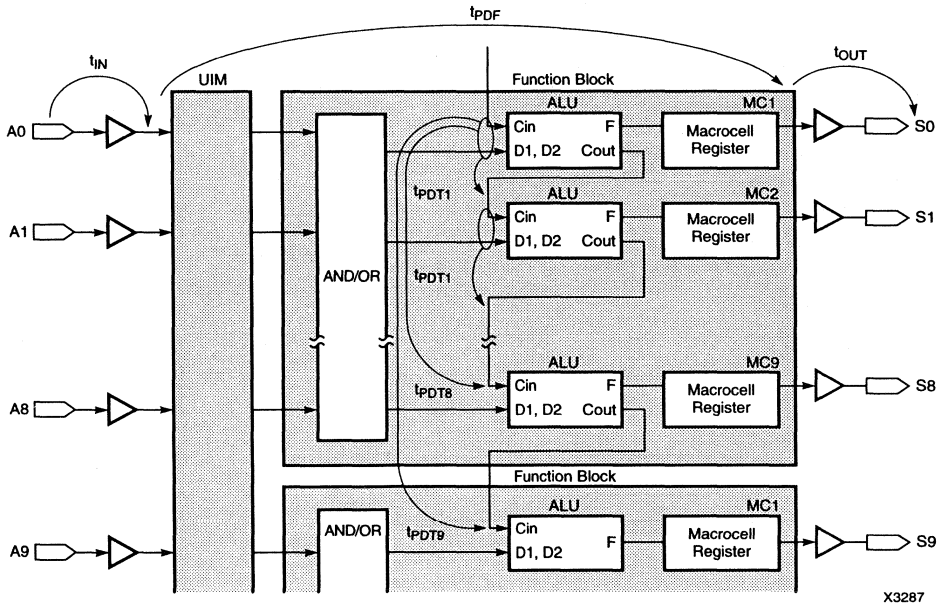


Figure 12. Arithmetic Timing Parameters

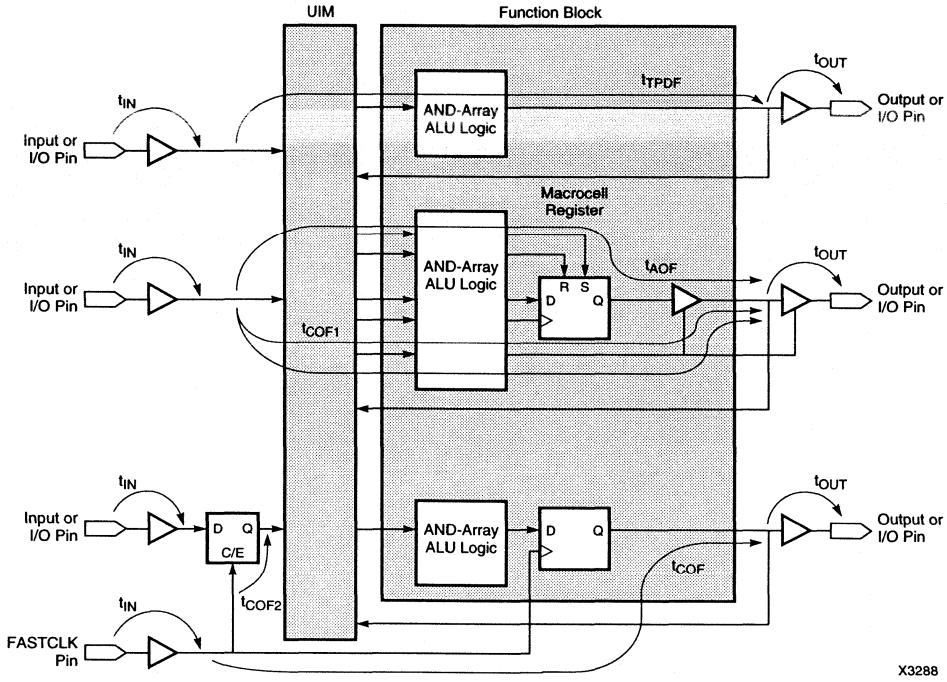


Figure 13. Incremental Timing Parameters

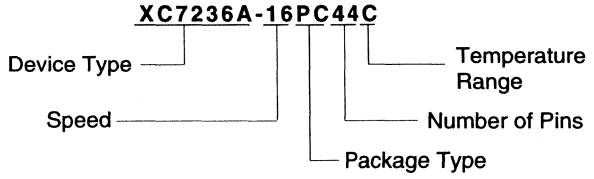
44-Pin LCC Pinouts

Pin #	Input	Output
1	Master Reset V_{PP}	
2	Input	MC2-1
3	Input	
4	Input	
5	Input	MC2-4
6	Input	MC2-5
7	GND	
8	Input	MC2-6
9	FastCLK0	MC2-7
10	FastCLK1	MC2-8
11	FastCLK2	MC2-9
12	V_{CCIO}	
13	Input	MC1-1
14	Input	MC1-2
15	Input	MC1-3
16	Input	MC1-4
17	GND	
18	Input	MC1-5
19	Input	MC1-6
20	Input/FI	MC1-7
21	Input/FI	MC1-8
22	Input/FI	MC1-9

Pin #	Input	Output
23	V_{CCIO}	
24	Input/FI	MC4-9
25	Input/FI	MC4-8
26	Input/FI	MC4-7
27	Input	MC4-6
28	Input	MC4-5
29	GND	
30	Input	MC4-4
31	Input	MC4-3
32	FastOE	MC4-2
33	Input	MC4-1
34	V_{CCINT}	
35	Input/FI	MC3-9
36	Input/FI	MC3-8
37	Input/FI	MC3-7
38	Input	MC3-6
39	GND	
40	Input	MC3-5
41	Input	MC3-4
42	Input	MC3-3
43	Input	MC3-2
44	Input	MC3-1

FI = Fast Input

Ordering Information



XC7236A -25 25 ns (40 MHz) sequential cycle time
-20 20 ns (50 MHz) sequential cycle time
-16 16 ns (60 MHz) sequential cycle time (commercial and industrial only)

Package Options

PC44 44-Pin Plastic Leaded Chip Carrier
WC44 44-Pin Windowed Ceramic Leaded Chip Carrier

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	-40°C to 85°C
M	Military	-55°C to 125°C (Case)

Component Availability

Pins	44		68		84		
Type	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Ceramic PGA
Code	PC44	WC44	PC68	WC68	PC84	WC84	PG84
XC7236A -25	CI	CI(M)					
XC7236A -20	CI	CI(M)					
XC7236A -16	CI	CI					

C = Commercial = 0° to +70°C I = Industrial = -40° to 85°C
Parenthesis indicate future product plans

X5273

XC7272A
72-Macrocell CMOS EPLD

Preliminary Product Specifications

Features

- Second-Generation High Density Programmable Logic Device
- UV-erasable CMOS EPROM technology
- 72 Macrocells, grouped into eight Function Blocks, interconnected by a programmable Universal Interconnect Matrix
- Each Function Block contains a programmable AND-array with 21 complementary inputs, providing up to 16 product terms per Macrocell
- Enhanced logic features:
  - 2-input Arithmetic Logic Unit in each Macrocell
  - Dedicated fast carry network between Macrocells
  - Wide AND capability in the Universal Interconnect Matrix
- Identical timing for all interconnect paths and for all Macrocell logic paths
- 72 signal pins in the 84-pin packages
  - 42 I/Os, 12 inputs, 18 outputs
- Each input is programmable
  - Direct, latched, or registered
- I/O-pin is usable as input when Macrocell is buried
- Two high-speed, low-skew global clock inputs
- 68-pin and 84-pin leaded chip carrier packages and 84-pin Pin-Grid-Array packages

General Description

The XC7272A is a second-generation High Density Programmable Logic Device that combines the classical features of the PAL-like EPLD architecture with innovative systems-oriented logic enhancements. This favors the implementation of fast state machines, large synchronous counters and fast arithmetic, as well as multi-level general-purpose logic. Performance, measured in achievable system clock rate and critical delays, is not only predictable, but independent of physical logic mapping, interconnect routing, and resource utilization. Performance, therefore, remains invariant between design iterations. The propagation delay through interconnect and logic is constant for any function implemented in any one of the output Macrocells.

The functional versatility of the traditional programmable logic array architecture is enhanced through additional gating and control functions available in an Arithmetic Logic Unit (ALU) in each Macrocell. Dedicated fast arithmetic carry lines running directly between adjacent Macrocells and Function Blocks support fast adders, subtractors and comparators of any length up to 72 bits.

This additional ALU in each Macrocell can generate any combinatorial function of two sums of products, and it can generate and propagate arithmetic-carry signals between adjacent Macrocells and Functional Blocks.

The Universal Interconnect Matrix (UIM) facilitates unrestricted, fixed-delay interconnects from all device inputs and Macrocell outputs to any Function Block AND-array input. The UIM can also perform a logical AND across any number of its incoming signals on the way to any Functional Block, adding another level of logic without additional delay. This supports bidirectional loadable synchronous counters of any size up to 72 bits, operating at the specified maximum device frequency

As a result of these logic enhancements, the XC7272A can deliver high performance even in designs that combine large numbers of product terms per output, or need more layers of logic than AND-OR, or need a wide AND function in some of the product terms, or perform wide arithmetic functions.

Automated design mapping and optimization is supported by Xilinx XEPLD development software based on design capture using third-party schematic entry tools, PLD compilers or direct text-based equation files. Design mapping is completed in a few minutes on a PC, or workstation.

Architectural Overview

Figure 1 shows the XC7272A structure. Eight Function Blocks (FBs) are all interconnected by a central UIM. Each FB receives 21 signals from the UIM and each FB produces nine signals back into the UIM. All device inputs are also routed via the UIM to all FBs. Each FB contains nine output Macrocells that draw from a programmable AND array driven by the 21 signals from the UIM. Most Macrocells drive a 3-state chip output, all feed back into the UIM.

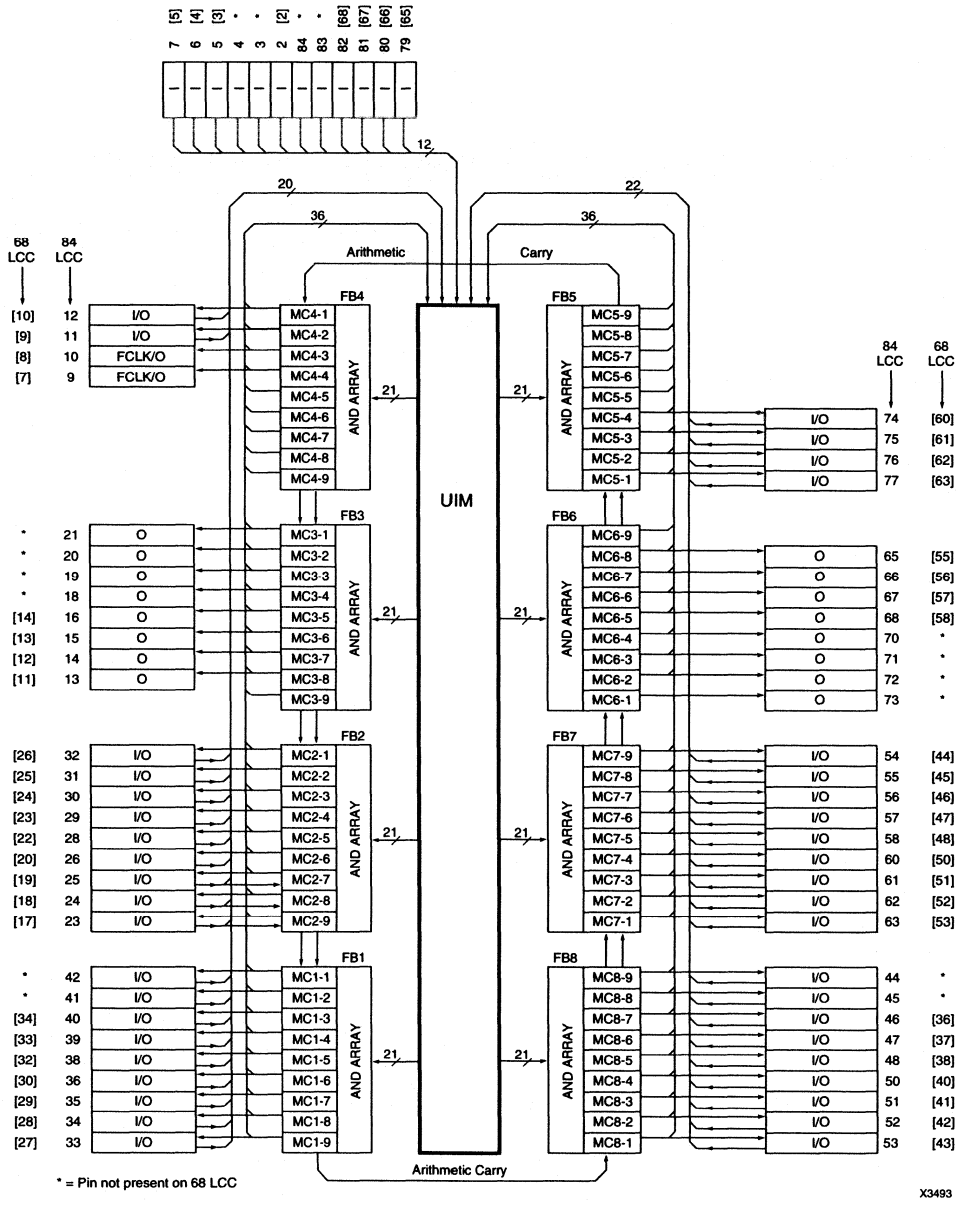


Figure 1. XC7272A Architecture

Function Blocks and Macrocells

The XC7272A contains 72 Macrocells with identical structure, grouped into eight Function Blocks of nine Macrocells each. Each Macrocell is driven by product terms derived from the 21 inputs from the UIM into the Function Block.

Five product terms are private to each Macrocell; an additional 12 product terms are shared among the nine Macrocells in any Function Block. One of the private product terms is a dedicated clock for the flip-flop in the Macrocell. See the description on page 3-24 for other clocking options.

The remaining four private product terms can be selectively ORed together with up to three of the shared product terms, and drive one input to an Arithmetic Logic Unit. The other input to the ALU is driven by the OR of up-to-nine product terms from the remaining shared product terms.

As a programmable option, two of the private product terms can be used for other purposes. One is the asynchronous active-High Reset of the Macrocell flip-flop, the other can be either an asynchronous active-High Set of the Macrocell flip-flop, or provide an active-High Output-Enable signal from any one of the Function Block inputs.

The Arithmetic Logic Unit has two programmable modes: In the *logic mode,* it is a 2-input function generator, a 4-bit look-up table, that can be programmed to generate any Boolean function of its two inputs. It can OR them, widening the OR function to max 16 inputs; it can AND them, which means that one sum of products can be used to mask the other; it can XOR them, toggling the flip-flop or comparing the two sums of products. Either or both of the sum-of-product inputs to the ALU can be inverted, and either or both can be ignored. The ALU can implement one additional layer of logic without any speed penalty.

In the *arithmetic mode,* the ALU block can be programmed to generate the arithmetic sum or difference of two operands, combined with a carry signal coming from the lower Macrocell; it also feeds a carry output to the next higher Macrocell. This carry propagation chain crosses the boundaries between Function Blocks, but it can also be configured 0 or 1 when it enters a Function Block.

This dedicated carry chain overcomes the inherent speed and density problems of the traditional EPLD architecture, when trying to perform arithmetic functions like add, subtract, and magnitude compare.

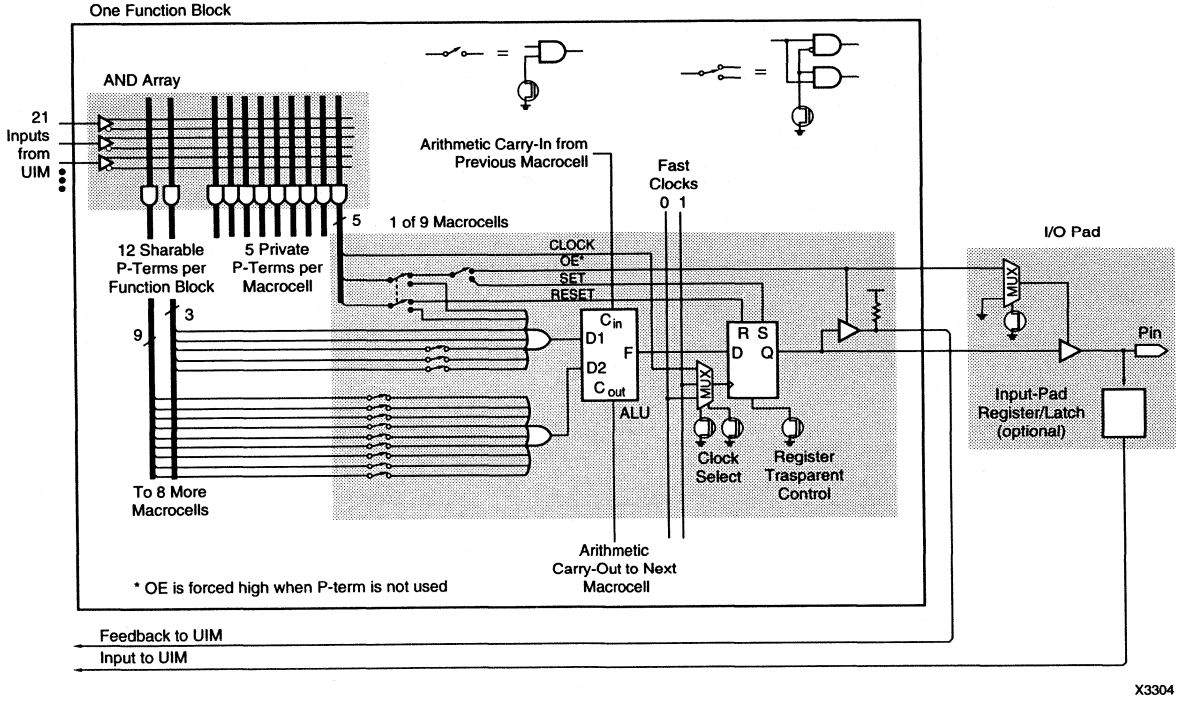


Figure 2. Function Block and Macrocell Schematic

The ALU output drives the D input of the Macrocell flip-flop.

Each flip-flop has several programmable options:

One option is to eliminate the flip-flop by making it transparent, which makes the Q output identical with the D input, independent of the clock.

If this option is *not* programmed, the flip-flop operates in the conventional manner, triggered by the rising edge on its clock input.

The clock source is programmable: It is either the dedicated product term mentioned above, or it is one of the two global FastCLK signals that are distributed with short delay and minimal skew over the whole chip.

The asynchronous Set and Reset (Clear) inputs override the clocked operation. If both asynchronous inputs are active simultaneously, Reset overrides Set. Upon power-up, each Macrocell flip-flop can be preloaded with either 0 or 1.

In addition to driving the chip output buffer, the Macrocell output is also routed back as an input to the UIM. When the Output Enable product term mentioned above is not active, this feedback line is forced High and thus disabled.

Universal Interconnect Matrix

The UIM receives 126 inputs: 72 from the 72 Macrocells, 42 from bidirectional I/O pins, and 12 from dedicated input pins. Acting as an unrestricted crossbar switch, the UIM generates 168 output signals, 21 to each Function Block.

Any one of the 126 inputs can be programmed to be connected to any number of the 168 outputs. The delay through the array is constant, independent of the apparent routing distance, the fan-out, fan-in, or routing complexity. Routability is not an issue: Any UIM input can drive any UIM output, even multiple outputs, and the delay is constant.

When multiple inputs are programmed to be connected to the same output, this output becomes the AND of the input signals if the levels are interpreted as active High. By choosing the appropriate signal inversion in the Macrocell outputs and the Function Block AND-array input, this AND-logic can also be used to implement a NAND, OR, or NOR function, thus offering an additional level of logic without any speed penalty.

A Macrocell feedback signal that is disabled by the output enable product term represents a High input to the UIM. Several such Macrocell outputs programmed onto the

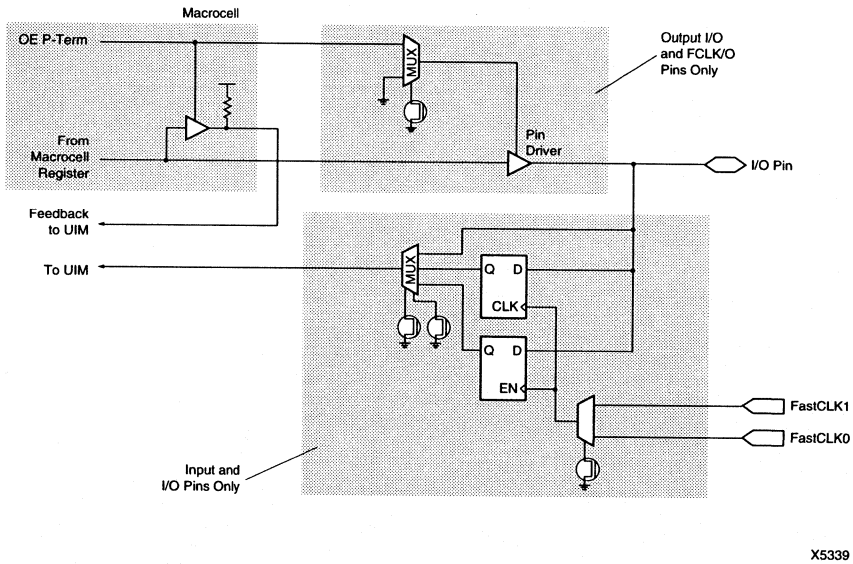


Figure 3. Input/Output Schematic

same UIM output thus emulating a 3-state bus line. If one of the Macrocell outputs is enabled, the UIM output assumes that same level.

Outputs

Sixty of the 72 Macrocells drive chip outputs directly through 3-state output buffers, each individually controlled by the Output Enable product term mentioned above. For bidirectional I/O pins, an additional programmable cell can optionally disable the output permanently. The buried flip-flop is then still available for internal feedback, and the pin can still be used as a separate input

Inputs

Each signal input to the chip is programmable as either direct, latched, or registered in a flip-flop. Latch and flip-flop can be programmed with either of the two FastCLK signals as latch enable or clock. The latch is transparent when FastCLK is High, and the flip-flop clocks on the rising edge of FastCLK. Registered inputs allow high system clock rates by pipelining the inputs before they incur the combinatorial delay in the device, in cases where a pipeline cycle is acceptable.

The direct, latched, or registered inputs then drive the UIM. There is no propagation-delay difference between pure inputs and I/O inputs.

Programming and Using the XC7272A

The features and capabilities described above are used by the Xilinx XEPLD development software to program the device according to the specification given either through schematic entry, or through a behavioral description expressed in Boolean equations.

The user can specify a security bit that prevents any reading of the programming bit map after the device has been programmed and verified.

The device is programmed in a manner similar to an EPROM (ultra-violet light erasable read-only memory) using the Intel Hex or JEDEC format. Programming support is available from a number of programmer manufacturers. The UIM connections and Function Block AND-array connections are made directly by non-volatile EPROM cells. Other control bits are read out of the EPROM array and stored into latches just after power-up. This method, common among EPLD devices, requires either a very fast V_{CC} rise time (<5 µs) or the application of a master-reset signal delayed at least until V_{CC} has reached the required operating voltage. The latter can be achieved using a simple capacitor and pull-up resistor on the MR pin (the RC product should be larger than twice the V_{CC} rise time). The power-up or reset signal initiates a self-timed configuration period lasting about 350 µs (t_{RESET}), during which all device outputs remain disabled and programmed preload state values are loaded into the macrocell registers.

Unused input and I/O pins should be tied to ground or Vcc or some valid logic level. This is common practice for all CMOS devices to avoid dissipating excess current through the input-pad circuitry.

The recommended decoupling capacitance on the three V_{CC} pins should total 1 µF using high-speed (tantalum or ceramic) capacitors.

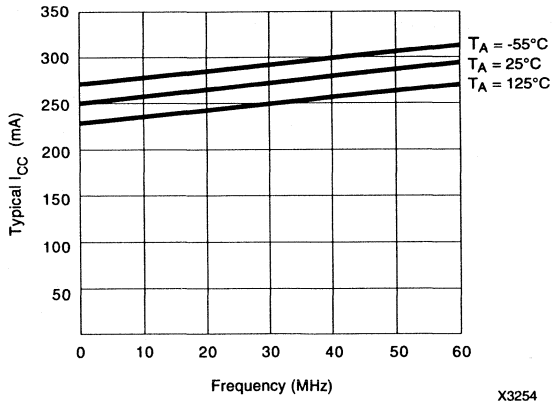


Figure 4. Typical I_{CC} vs Frequency for XC7272A Configured as Sixteen 4-bit Counters
(V_{CC} = +5.0 V, V_{IN} = V_{CC} or GND, all outputs open)

Absolute Maximum Ratings

			Units
V_{CC}	Supply voltage relative to GND	–0.5 to 7.0	V
V_{IN}	Input voltage with respect to GND	–0.5 to 7.0	V
V_{TS}	Voltage applied to 3-state output	–0.5 to 7.0	V
T_{STG}	Storage temperature	–65 to + 150	°C
T_{SOL}	Maximum soldering temperature (10 s @ 1/16 in. = 1.5 mm)	+ 260	°C

Note: Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those listed under Operating Conditions is not implied. Exposure to Absolute Maximum Ratings conditions for extended periods of time may affect device reliability.

Operating Conditions

		Min	Max	Units
V_{CC}	Supply voltage relative to GND Commercial T_A = 0°C to 70°C	4.75	5.25	V
	Supply voltage relative to GND Industrial T_A = –40°C to 85°C	4.5	5.5	V
	Supply voltage relative to GND Military T_C = –55°C to 125°C	4.5	5.5	V
V_{IH}	High-level input voltage	2.0	V_{CC}+ 0.3	V
V_{IL}	Low-level input voltage	0	0.8	V

DC Characteristics Over Operating Conditions

		Min	Max	Units
V_{OH}	High-level output voltage @ I_{OH} = –4 mA , V_{CC} min	2.4		V
V_{OL}	Low-level output voltage @ I_{OL} = 8 mA , V_{CC} min		0.5	V
I_{CC}	Supply current V_{IN} = 0 V, V_{CC} = Max, f = 0 MHz	252 mA Typ		
I_{IL}	Input Leakage current	–10	+10	µA
I_{OZ}	Output High-Z leakage current	–100	+100	µA
C_{IN}	Input capacitance (sample tested)		10	pF

AC Timing Requirements

Symbol	Parameter	Fig.	XC7272A-25		XC7272A-20		*XC7272A-16		Units
			Min	Max	Min	Max	Min	Max	
f_{CYC} (Note 1)	Max sequential toggle frequency (with feedback) using FastCLK	6	40		50		60		MHz
f_{CYC1} (Note 1)	Max sequential toggle frequency (with feedback) using a Product-Term clock	6	40		50		60		MHz
f_{CLK} (Note 5)	Max Macrocell register transmission frequency (without feedback) using FastCLK		59		60		60		MHz
f_{CLK1} (Note 5)	Max Macrocell register transmission frequency (without feedback) using a Product-Term clock		50		50		60		MHz
f_{CLK2} (Note 5)	Max input register transmission frequency (without feedback) using FastCLK		67		67		67		MHz
f_{CLK3} (Note 1)	Max input register to Macrocell register pipeline frequency using FastCLK	7	40		50		60		MHz
t_{WL}	FastCLK Low pulse width	11	7.5		7.5		6		ns
t_{WH}	FastCLK High pulse width	11	7.5		7.5		6		ns
t_{W1}	Product-Term clock pulse width (active/inactive)	11	10		9		7		ns
t_{SU}	Input to Macrocell register set-up time before FastCLK	9	24		19		15		ns
t_{H}	Input to Macrocell register hold time after FastCLK	9	-7		-4		-4		ns
t_{SU1} (Note 1)	Input to Macrocell register set-up time before Product-Term clock	8	10		8		6		ns
t_{H1}	Input to Macrocell register hold time after Product-Term clock	8	0		0		0		ns
t_{SU2}	Input to register/latch set-up time before FastCLK	10	8		8		6		ns
t_{H2}	Input to register/latch set-up time after FastCLK	10	0		0		0		ns
t_{WA}	Set/Reset pulse width	11	12		10		8		ns
t_{RA}	Set/Reset input recovery set-up time before FastCLK	11	20		20		16		ns
t_{HA}	Set/Reset input hold time after FastCLK	11	-5		-3		-3		ns
t_{RA1}	Set/Reset input recovery time before P-Term clock	11	6		5		4		ns
t_{HA1}	Set/Reset input hold time after P-Term clock	11	9		8		6		ns
t_{HRS}	Set/Reset input hold time after Reset/Set inactive		10		8		6		ns

*Commercial/Industrial Only

X5211

Propagation Delays

Symbol	Parameter	Fig.	XC7272A-25		XC7272A-20		*XC7272A-16		Units
			Min	Max	Min	Max	Min	Max	
t_{CO}	FastCLK input to registered output delay	11	5	16	3	14	3	12	ns
t_{CO1}	P-Term clock input to registered output delay	11	10	30	6	25	6	21	ns
t_{AO}	Set/Reset input to registered output delay	11	13	40	8	32	8	25	ns
t_{PDD} (Note 1)	Input to nonregistered output delay	11	13	40	8	32	8	25	ns
t_{OE}	Input to output enable	11	11	32	7	25	7	22	ns
t_{OD}	Input to output disable		11	32	7	25	7	22	ns

*Commercial/Industrial Only

X5212

Notes 1. Specifications account for logic paths which use the maximum number of available product terms and the ALU.

Incremental Parameters

Symbol	Parameter	Fig.	XC7236A-25		XC7236A-20		*XC7236A-16		Units
			Min	Max	Min	Max	Min	Max	
t_{PDT1} (Note 2)	Arithmetic carry delay between adjacent Macrocells	12		1.6		1.2		1	ns
t_{PDT8} (Note 2)	Arithmetic carry delay through 9 adjacent Macrocells in a Function Block	12		10		8		6	ns
t_{PDT9} (Note 2)	Arithmetic carry delay through 10 Macrocells from Macrocell #n to Macrocell #n in next F Block	12		14		12		10	ns
t_{COF}	Incremental delay from FastCLK net to registered output feedback	13		1		1		1	ns
t_{COF1}	Incremental delay from UIM-input (for P-Term clock) to registered Macrocell feedback	13		1.5		12		10	ns
t_{COF2} (Note 3)	Incremental delay from FastCLK net to latched/registered UIM-input	13		1		1		1	ns
t_{PDF} (Note 1)	Incremental delay from UIM-input to nonregistered Macrocell feedback	13		25		19		14	ns
t_{AOF}	Incremental delay from UIM-input (Set/Reset) to registered Macrocell feedback	13		25		19		14	ns
t_{OEF}, t_{ODF}	Incremental delay from UIM-input (used as output-enable/disable) to Macrocell feedback	13		17		12		11	ns
t_{IN} + t_{OUT} (Note 4)	Propagation delay through unregistered input pad (to UIM) plus output pad driver (from Macrocell)	13		15		13		11	ns

*Commercial/Industrial Only

X5213

Power-up/Reset Timing Parameters

Symbol	Description	Min	Typ	Max	Units
t_{WMR}	Master Reset input Low pulse width	100			ns
t_{rVCC}	V_{CC} rise time (if MR not used for power-up)			5	µs
t_{RESET}	Configuration completion time (to outputs operational)		350	1000	µs

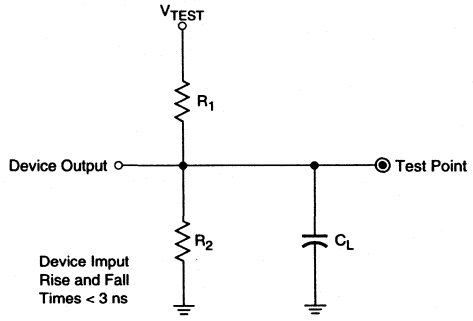
Notes 1. Specifications account for logic paths which use the maximum number of available product terms and the ALU.

2. Arithmetic carry delays are measured as the increase in required set-up time to adjacent Macrocell(s) for an adder with registered outputs.

3. Parameter t_{COF2} is derived as the difference between the clock period for pipelining input-to-Macrocell registers ($1/f_{CLK3}$) and the non-registered input set-up time (t_{SU}).

4. Parameter t_{IN} represents the delay from an input or I/O pin to a UIM-input (or from a FastCLK pin to the Fast CLK net); t_{OUT} represents the delay from a Macrocell output (feedback point) to an output or I/O pin. Only the sum of t_{IN} + t_{OUT} can be derived from measurements, e.g., $t_{IN} + t_{OUT} = t_{SU} + t_{CO} - 1/f_{CYC}$.

5. Not tested but derived from appropriate pulse-widths, set-up time and hold-time measurements.



Output Type	V_{CCIO}	V_{TEST}	R_1	R_2	C_L
O	5.0 V	5.0 V	450 Ω	245 Ω	35 pF

X3490

Figure 5. AC Load Circuit

Timing and Delay Path Specifications

The delay path consists of three blocks that can be connected in series:

- Input Buffer and associated latch or register
- Logic Resource (UIM, AND-array and Macrocell)
- Three-state Output Buffer

All inputs have the same delay, regardless of fan-out or location. All logic resources have the same delay, regardless of logic complexity, interconnect topology or location on the chip. All outputs have the same delay. The achievable clock rate is, therefore, determined only by the input method (direct, latched or registered) and the number of times a signal passes through the combinatorial logic.

Timing and Delay Path Descriptions

Figure 6 defines the max clock frequency (with feedback). Any Macrocell output can be fed back to the UIM as an input for the next clock cycle. Figure 6 shows the relevant delay path. The parameters f_{CYC} and f_{CYC1} specify the maximum operating frequency for FastCLK and product-term clock operation respectively.

Figure 7 specifies the max operating frequency (f_{CLK3}) for pipelined operation between the input registers and the Macrocell registers, using FastCLK.

Figure 8 defines the set-up and hold times from the data inputs to the product-term clock used by the output register.

Figure 9 defines the set-up and hold times from the data inputs to the FastCLK used by the output register.

Figure 10 defines the set-up and hold times from the data input to the FastCLK used in an input register.

Figure 11 shows the waveforms for the Macrocell and control paths

Figure 12 defines the carry propagation delays between Macrocells and between Function Blocks. The parameters describe the delay from the C_{IN}, D1 and D2 inputs of a Macrocell ALU to the C_{IN} input of the adjacent Macrocell ALU. These delays must be added to the standard Macrocell delay path (t_{PD} or t_{SU}) to determine the performance of an arithmetic function.

Figure 13 defines the incremental parameters for the standard Macrocell logic paths. These incremental parameters are used in conjunction with pin-to-pin parameters when calculating compound logic path timing. Incremental parameters are derived indirectly from other pin-to-pin measurement.

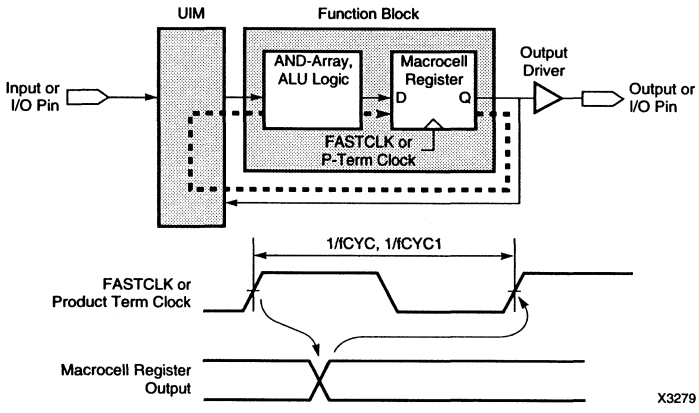


Figure 6. Delay Path Specifications for f_{CYC} and f_{CYC1}

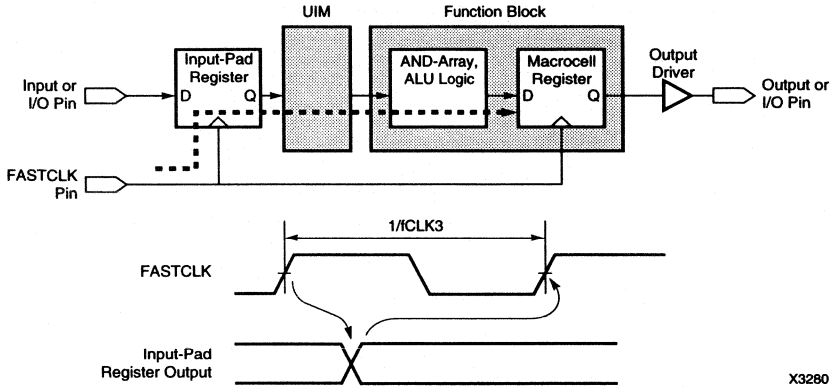


Figure 7. Delay Path Specification for f_{CLK3}

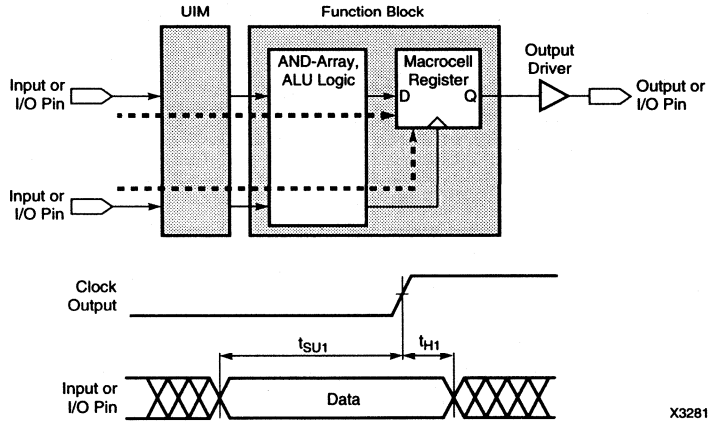


Figure 8. Delay Path Specification for t_{SU1} and t_{H1}

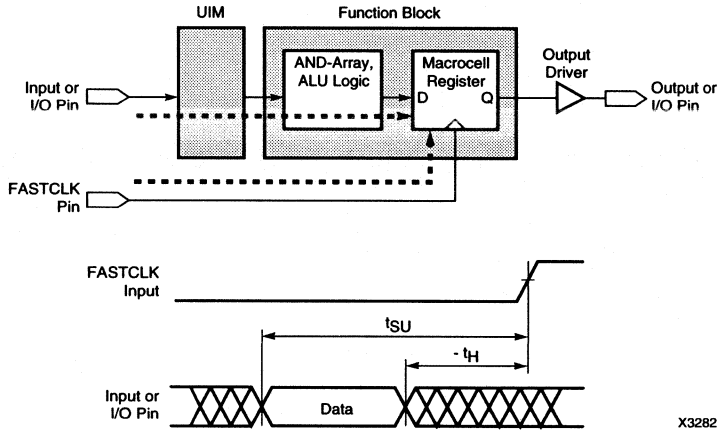


Figure 9. Delay Path Specification for t_{SU} and t_H

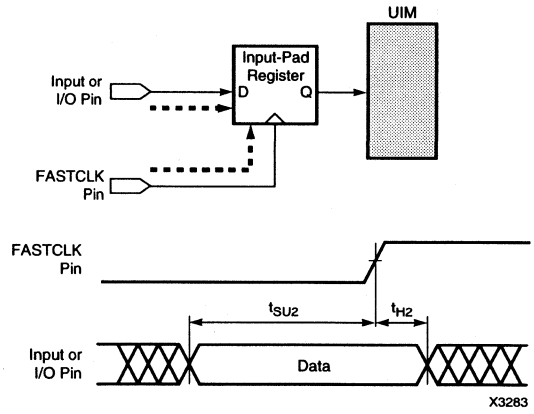


Figure 10. Delay Path Specification for t_{SU2} and t_{H2}

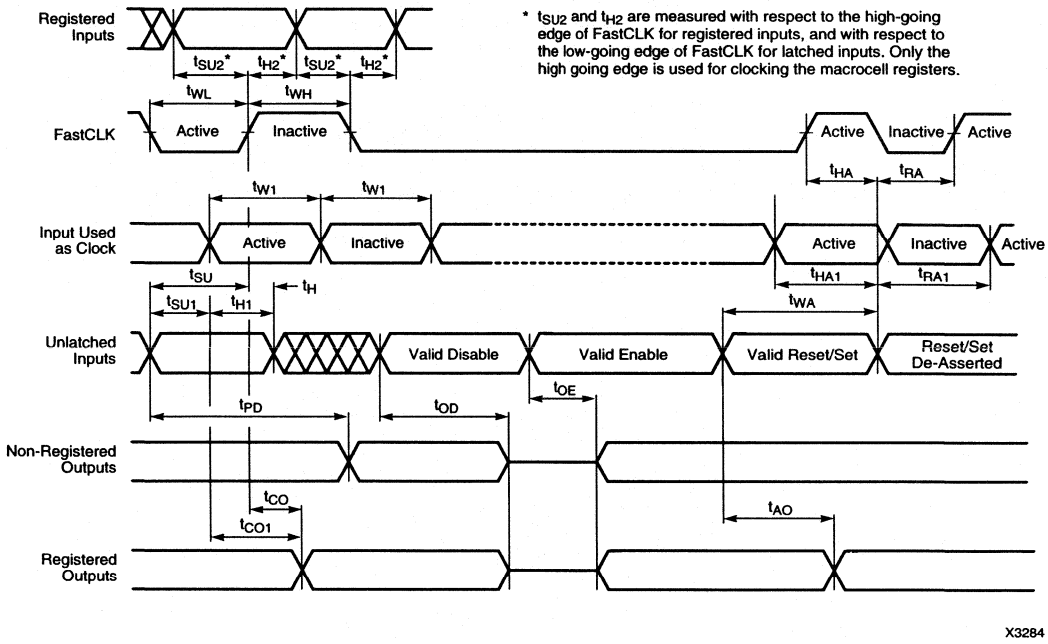


Figure 11. Principal Pin-to-Pin Measurements

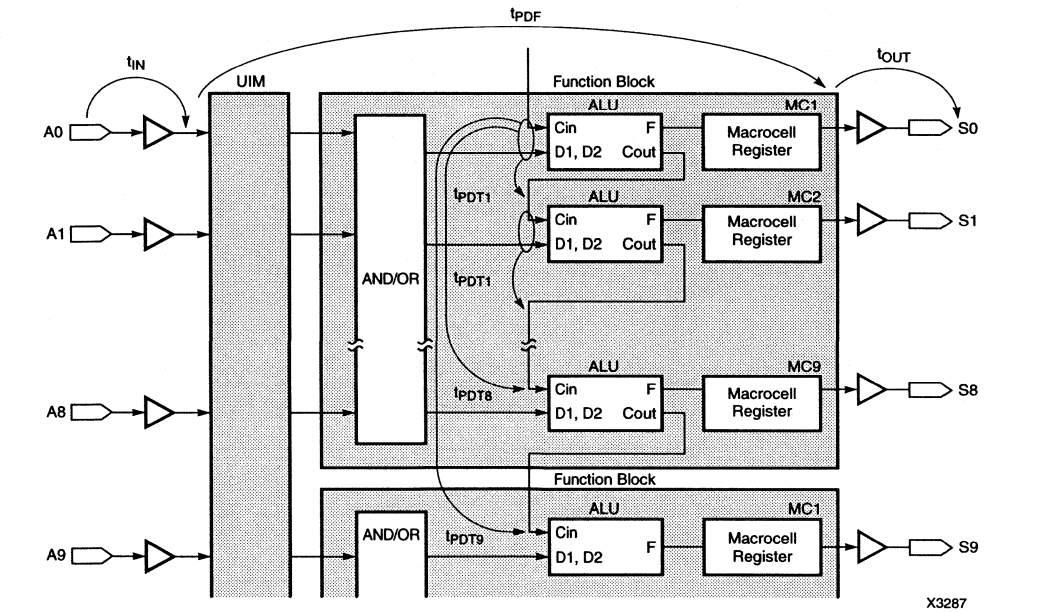


Figure 12. Arithmetic Timing Parameters

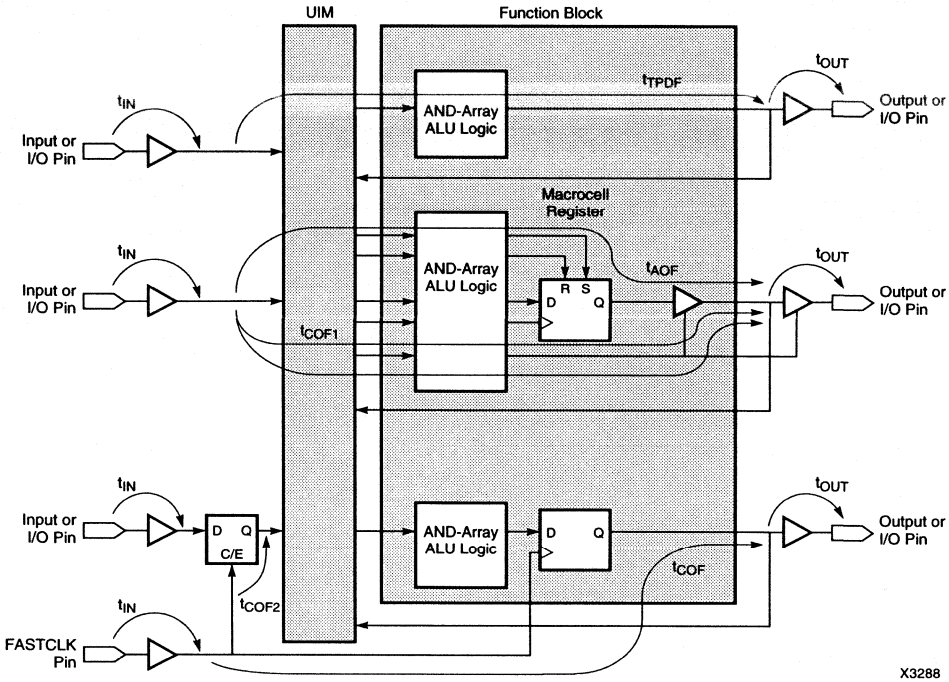


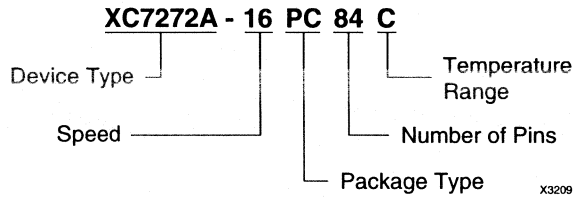
Figure 13. Incremental Timing Parameters

68-Pin LCC, 84-Pin LCC and PGA Pinouts

68 LCC	in XC7272A out		84 LCC	84 PGA
1	Master Reset		1	F-9
2	Input		2	F-11
-	Input		3	E-11
-	Input		4	E-10
3	Input		5	E-9
4	Input		6	D-11
5	Input		7	D-10
6	**GROUND**		8	C-11
7	Fast CLK0	MC4-4	9	B-11
8	Fast CLK1	MC4-3	10	C-10
9	Input	MC4-2	11	A-11
10	Input	MC4-1	12	B-10
11		MC3-8	13	B-9
12		MC3-7	14	A-10
13		MC3-6	15	A-9
14		MC3-5	16	B-8
15	**GROUND**		17	A-8
-		MC3-4	18	B-6
-		MC3-3	19	B-7
-		MC3-2	20	A-7
-		MC3-1	21	C-7
16	**Vcc**		22	C-6
17	Input	MC2-9	23	A-6
18	Input	MC2-8	24	A-5
19	Input	MC2-7	25	B-5
20	Input	MC2-6	26	C-5
21	**GROUND**		27	A-4
22	Input	MC2-5	28	B-4
23	Input	MC2-4	29	A-3
24	Input	MC2-3	30	A-2
25	Input	MC2-2	31	B-3
26	Input	MC2-1	32	A-1
27	Input	MC1-9	33	B-2
28	Input	MC1-8	34	C-2
29	Input	MC1-7	35	B-1
30	Input	MC1-6	36	C-1
31	**GROUND**		37	D-2
32	Input	MC1-5	38	D-1
33	Input	MC1-4	39	E-3
34	Input	MC1-3	40	E-2
-	Input	MC1-2	41	E-1
-	Input	MC1-1	42	F-2

68 LCC	in XC7272A out		84 LCC	84 PGA
35	**Vcc**		43	F-3
-	Input	MC8-9	44	G-3
-	Input	MC8-8	45	G-1
36	Input	MC8-7	46	G-2
37	Input	MC8-6	47	F-1
38	Input	MC8-5	48	H-1
39	**GROUND**		49	H-2
40	Input	MC8-4	50	J-1
41	Input	MC8-3	51	K-1
42	Input	MC8-2	52	J-2
43	Input	MC8-1	53	L-1
44	Input	MC7-9	54	K-2
45	Input	MC7-8	55	K-3
46	Input	MC7-7	56	L-2
47	Input	MC7-6	57	L-3
48	Input	MC7-5	58	K-4
49	**GROUND**		59	L-4
50	Input	MC7-4	60	J-5
51	Input	MC7-3	61	K-5
52	Input	MC7-2	62	L-5
53	Input	MC7-1	63	K-6
54	**Vcc**		64	J-6
55		MC6-8	65	J-7
56		MC6-7	66	L-7
57		MC6-6	67	K-7
58		MC6-5	68	L-6
59	**GROUND**		69	L-8
-		MC6-4	70	K-8
-		MC6-3	71	L-9
-		MC6-2	72	L-10
-		MC6-1	73	K-9
60	Input	MC5-4	74	L-11
61	Input	MC5-3	75	K-10
62	Input	MC5-2	76	J-10
63	Input	MC5-1	77	K-11
64	**GROUND**		78	J-11
65	Input		79	H-10
66	Input		80	H-11
67	Input		81	F-10
68	Input		82	G-10
-	Input		83	G-11
-	Input		84	G-9

Ordering Information



Speed Options

-25 25 ns (40 MHz) sequential cycle time
-20 20 ns (50 MHz) sequential cycle time
-16 16 ns (60 MHz) sequential cycle time (commerial and industrial only)

Package Options

PC68 68-Pin Plastic Leaded Chip Carrier
WC68 68-Pin Windowed Ceramic Leaded Chip Carrier
PC84 84-Pin Plastic Leaded Chip Carrier
WC84 84-Pin Windowed Ceramic Leaded Chip Carrier
PG84 84-Pin Ceramic Windowed Pin Grid Array

Temperature Options

C	Commercial	0°C to 70°C
I	Industrial	-40°C to 85°C
M	Military	-55°C to 125°C (Case)

Component Availability

Pins	44		68		84		
Type	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Plastic PLCC	Ceramic CLCC	Ceramic PGA
Code	PC44	WC44	PC68	WC68	PC84	WC84	PG84
-25			CI	CI	CI	CI(M)	CI
XC7272A-20			CI	CI	CI	CI	CI
-16			CI	CI	CI	CI	CI

C = Commercial = 0° to +70°C I = Industrial = -40° to 85°C
Parenthesis indicate future product plans

X5274

XILINX®

Packages and Thermal Characteristics

Packages and Thermal Characteristics

Package Options

	Surface Mount						Through-hole
	PLCC	**PQFP**	**TQFP**	**VQFP**	**CQFP**	**BGA**	**PGA**
Standard	JEDEC	EIAJ	EIAJ	EIAJ	JEDEC	JEDEC	JEDEC
Lead Pitch	50 mil	0.65/0.5 mm	0.5 mm	0.5 mm	25 mil	1.5 mm	100 mil
Body	Plastic	Plastic	Plastic	Plastic	Ceramic	FR4	Ceramic/Plastic
Temp Options	C, I	C, I	C, I	C, I	M, B	C	C, I, M, B
Ordering Code	PC	PQ	TQ	VQ	CB	WB, PB	PG, PP
XC7236A	44						
XC7272A	68, 84						84
XC7336	44		100	100			
XC7354	44, 68						
XC7372	68, 84						
XC3108	84	160				225	144
XC73144						225	184
XC2064	44, 68						68
XC2018	44, 68, 84		100	100			84
XC3020/XC3120	68, 84	100			100		84
XC3030/XC3130	44, 68, 84	100	100	100			84
XC3042/XC3142	84	100	100	100	100		84, 132
XC3064/XC3164	84	160					132
XC3090/XC3190	84	160, 208			164		175
XC3195	84	160, 208					175, 223
XC4002A	84	100	100		100		120
XC4003/XC4003A	84	100	100				120
XC4003H		208					191
XC4004A	84	160	144				120
XC4005/XC4005A	84	160, 208	144		164		156
XC4005H		240					223
XC4006		160, 208					156
XC4008		208			196		191
XC4010/XC4010D	84	160, 208			196	225	191
XC4013		208, 240					223
XC4025		208				225	299

X3469

Inches vs. Millimeters

The standards for PLCC, CQFP and PGA packages were set by an American institution, JEDEC, which defined their dimensions in inches, with lead pitches of 25, 50 or 100 mils (i.e.; 0.025", 0.050" or 0.100").

The Japanese established the standard for PQFP, MQFP, TQFP and VQFP packages, and used metric units. Those devices have a lead pitch of 0.5 mm, except for the 100 and 160 PQFP, which have a lead pitch of 0.65 mm.

By definition, the lead pitch has no **accumulating** tolerance or error. However, that assumes that the values are described in their original measuring system. The derived values of 25.6 mil (for 0.65 mm) and 20 or 19.7 mil (for 0.5 mm) are not exact, leading to unacceptable errors when concatenated.

Because of the potential for measurement discrepancies, this Data Book provides measurements only in the controlling standard, either inches or millimeters. (The inch is officially defined as equal to 25.4 mm exactly).

Number of Available I/O Pins

	Max	Number of Package Pins																			
	I/O	44	64	68	84	100	120	132	144	156	160	164	175	176	184	191	196	208	223	225	240	299
XC7236A	36	36																				
XC7272A	72			56	72																	
XC7336	38	38																				
XC7354	58	38		58																		
XC7372	72			56	72																	
XC73108	120				72				120		120		120		120					120		
XC73144	156														156					156		
XC2064	58	34		58																		
XC2018	74	34		64	74	74																
XC3020/XC3120	64		54	58	64	64																
XC3030/XC3130	80	34	54	58	74	80																
XC3042/XC3142	96				74	82		96														
XC3064/XC3164	120				70			110			120											
XC3090/XC3190	144				70						138	142	144	144				144				
XC3195	176				70						138		144					176	176			
XC4002A	64				61	64	64															
XC4003/XC4003A	80				61	77	80															
XC4003H	160															160		160				
XC4004A	96				61		95		96		96											
XC4005/XC4005A	112				61				112	112	112	112						112				
XC4005H	192																		192		192	
XC4006	128									125	128							128				
XC4008	144															144	144	144				
XC4010/XC4010D	160				61											160	160	160		160		
XC4013	192																	160	192	192	192	
XC4025	256																		192		193	256

X3406

Cavity Up or Cavity Down

Most Xilinx devices attach the die against the inside bottom of the package, i.e. against the surface that, on the outside, does not carry the Xilinx logo and device designation. This is called "cavity-up", and has been the standard IC assembly method for over 25 years, although it is not ideal from a thermal point of view.

Large Xilinx Pin Grid Arrays (>130 pins) and **Ceramic** QuadFlatPaks are assembled differently. Here, the die is attached to the inside top of the package. Attaching the die against the inside top of the package optimizes heat transfer to the ambient air.

For most packages, this information may be of only academic interest, because the user has no choice. For Ceramic Quad Flat Pak (CQFP) packages, however, it may influence the way the user forms the leads. For best heat transfer to the surrounding air, these packages should be attached to the PC board in such a way that the surface with the logo faces up, away from the PC board, and the metal lid faces the PC board.

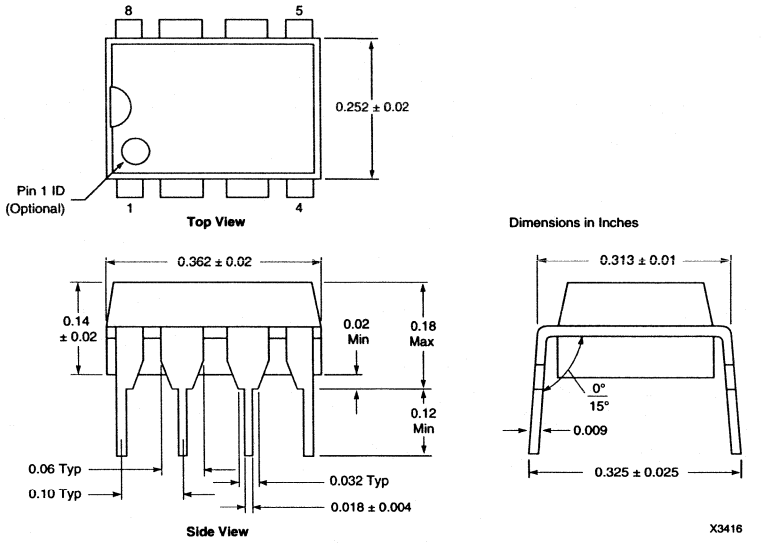
Clockwise or Counterclockwise

Obviously, the orientation of the die in the package and the orientation of the package on the PC board, both have a profound impact on the PC board layout.

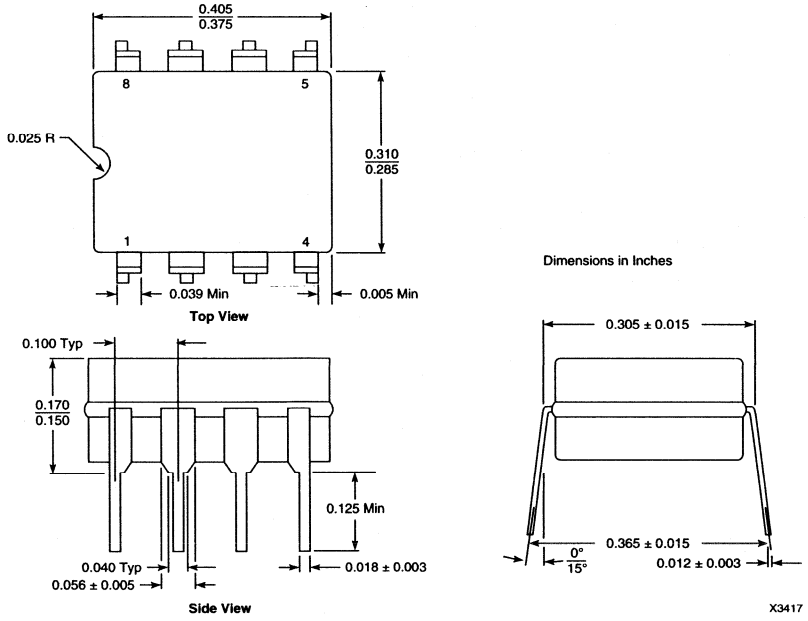
PLCC and PQFPs count pins in a **counterclockwise** direction, when viewed from the top of the package, the surface with the Xilinx logo. PLCCs have pin #1 in the center of the beveled edge, while all other packages have pin #1 in one corner. The exceptions are the 100- and 164-pin CQFPs (CB100 and CB164), where the XC4000 devices have pin #1 in a corner, but the XC3000 devices have pin #1 in the center of one edge.

CQFPs count pins in a **clockwise** direction, when viewed from the surface with the Xilinx logo. The user can make the pin count run in the more conventional counterclockwise direction, but that means forming the leads such that the logo mounts against the PC board, and heat flow to the surrounding air is impaired.

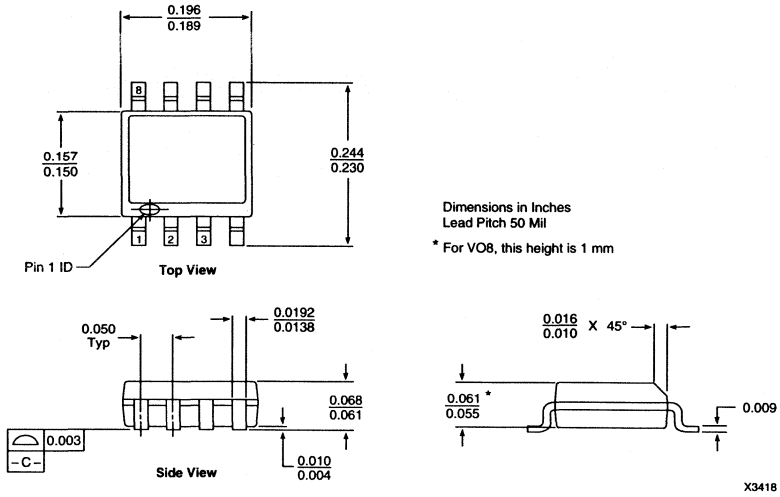
Physical Dimensions



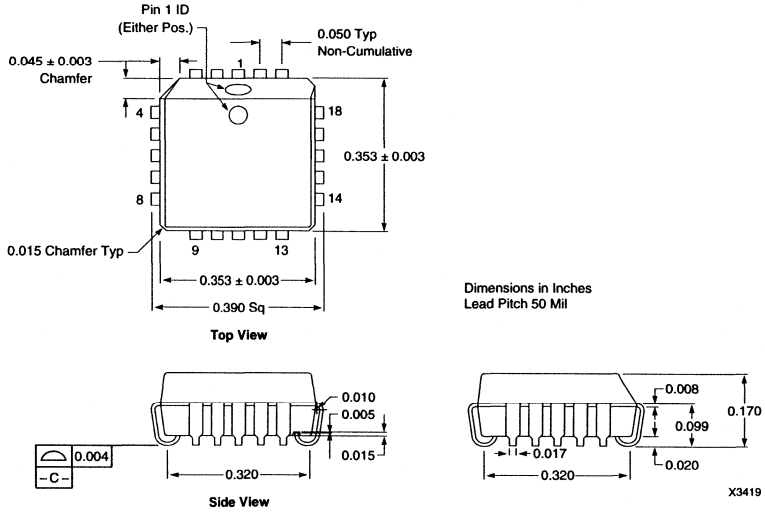
8-Pin Plastic DIP (PD8)



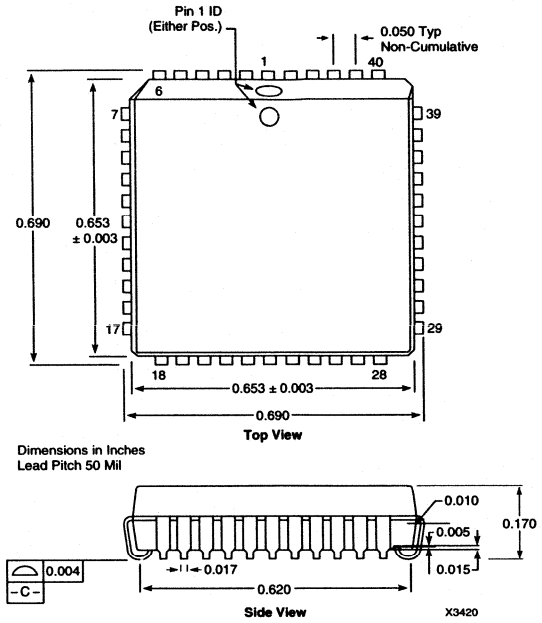
8-Pin Ceramic DIP (DD8)



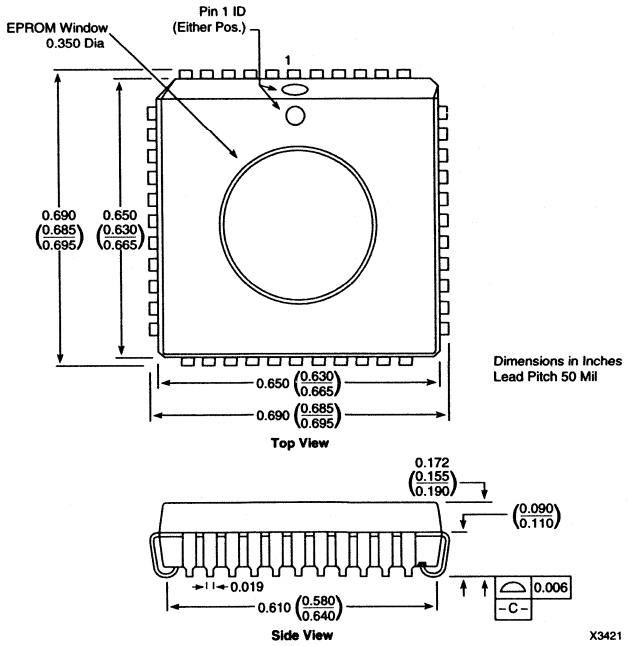
8-Pin Plastic Small Outline (SO8) and 8-Pin Plastic Small Outline Thin (VO8) Packages



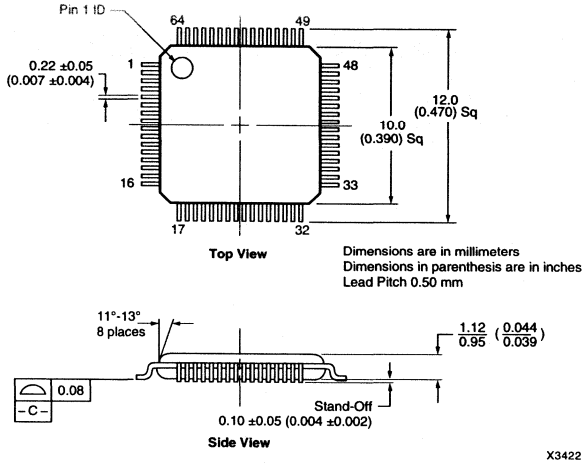
20-Pin Plastic PLCC (PC20)



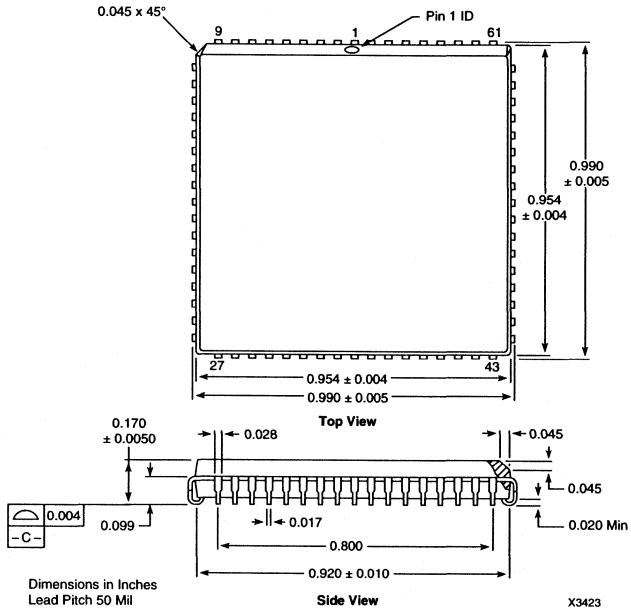
44-Pin Plastic PLCC (PC44)



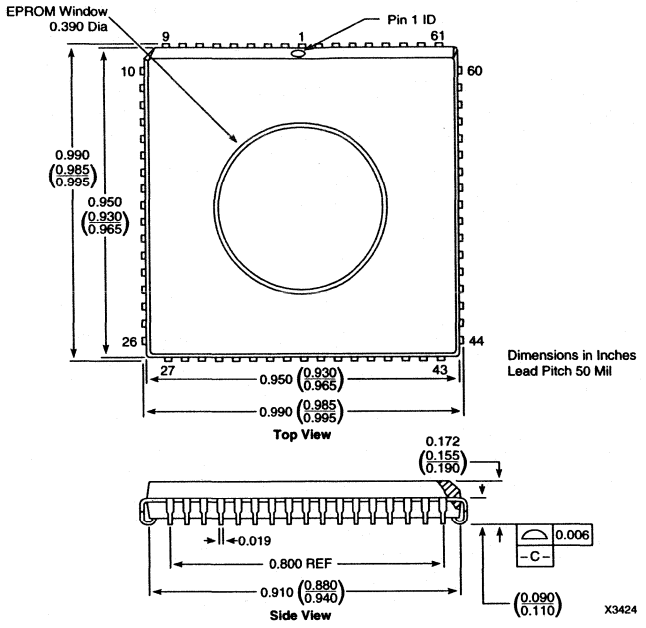
44-Pin Windowed Ceramic CLCC (WC44)



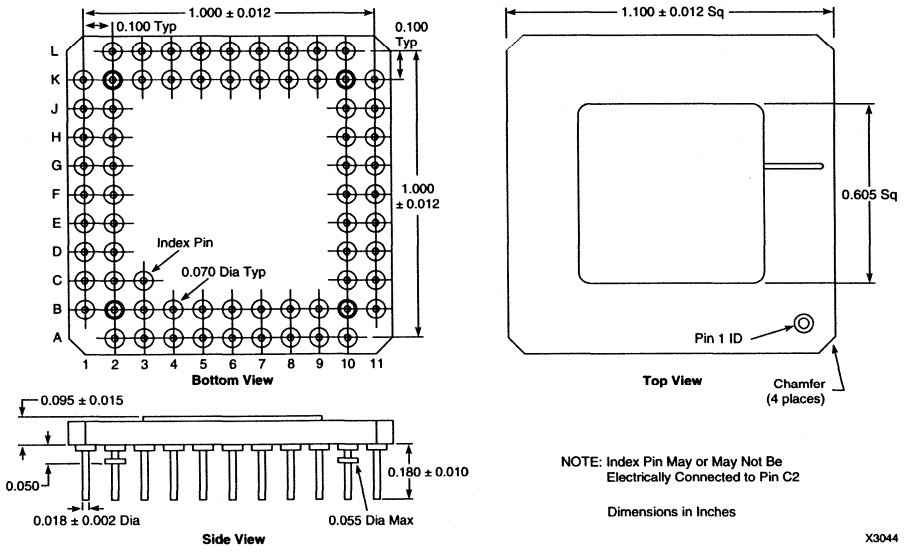
64-Pin Plastic VQFP (VQ64)



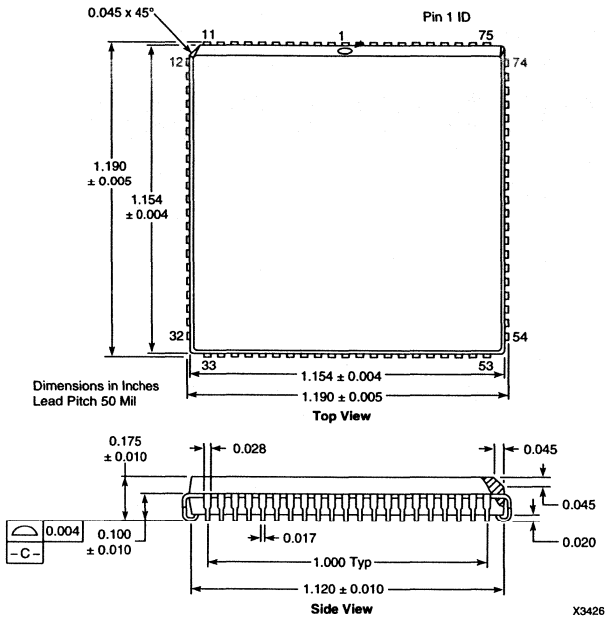
68-Pin Plastic PLCC (PC68)



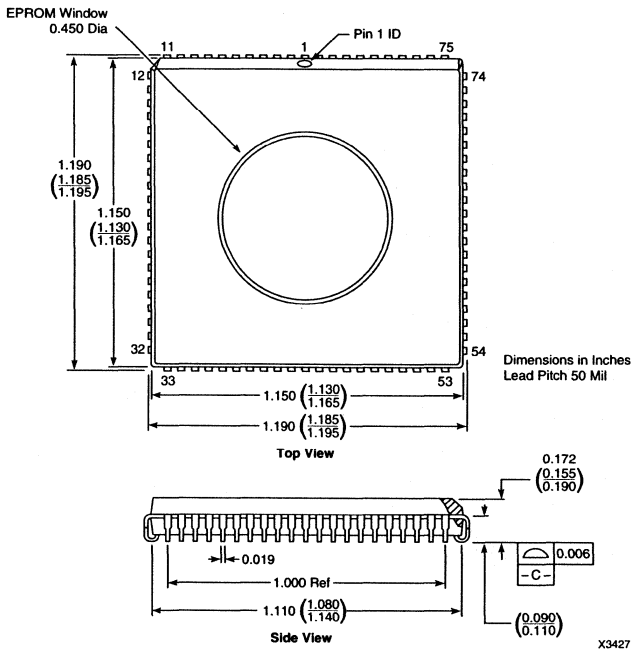
68-Pin Windowed Ceramic CLCC (WC68)



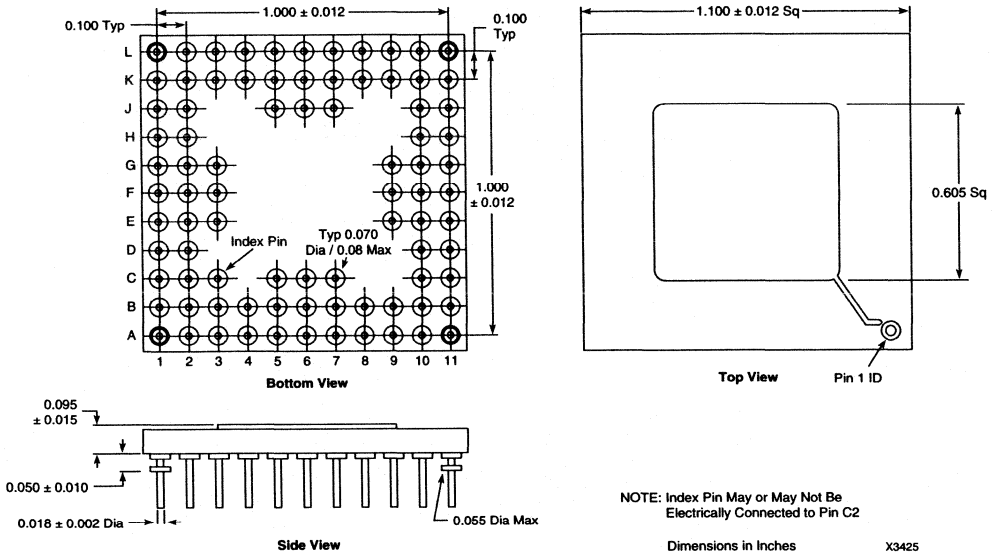
68-Pin Ceramic PGA (PG68)



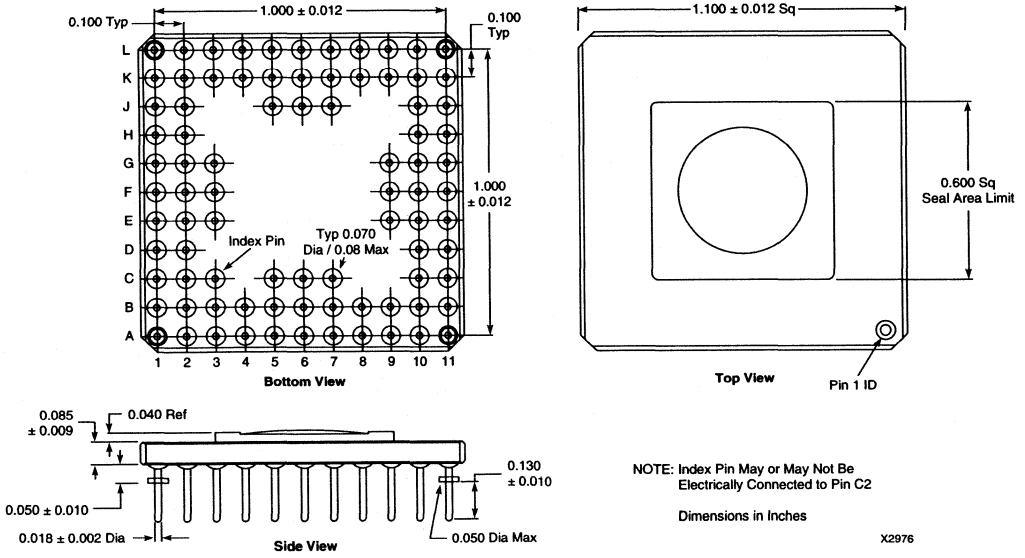
84-Pin Plastic PLCC (PC84)



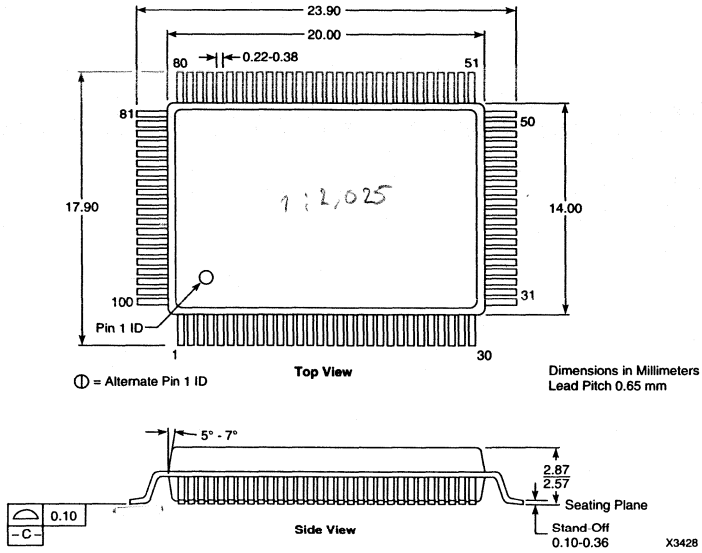
84-Pin Windowed Ceramic CLCC (WC84)



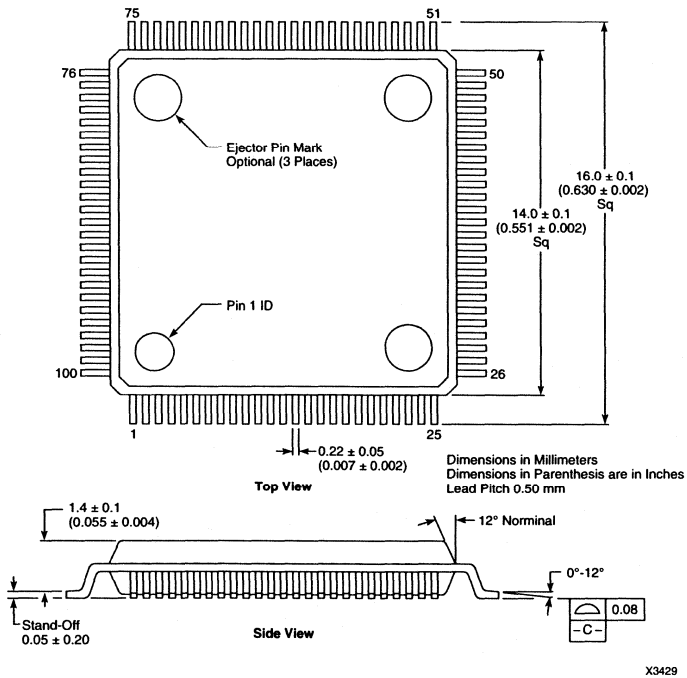
84-Pin Ceramic PGA (PG84)



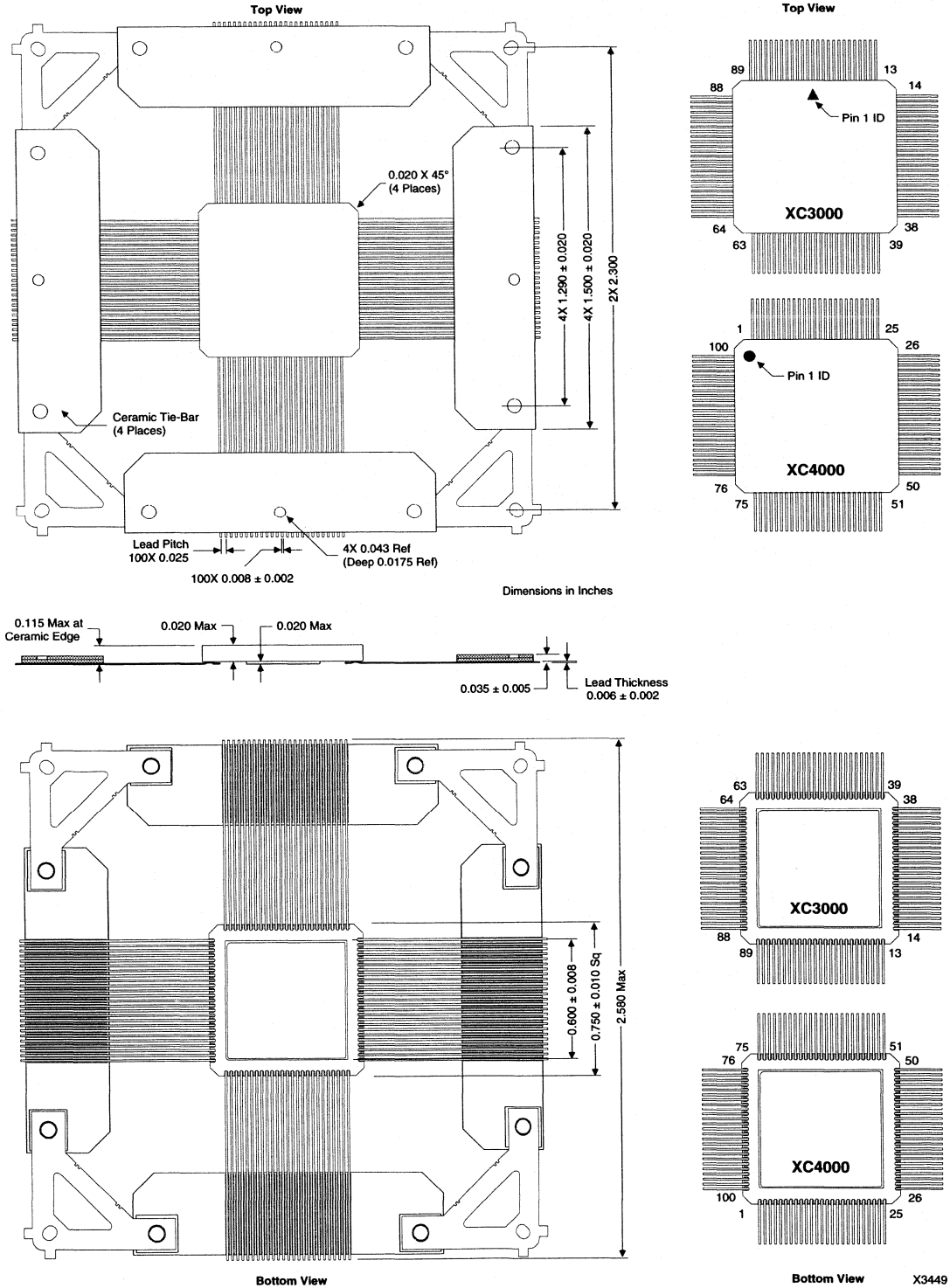
84-Pin Windowed Ceramic PGA (PG84)



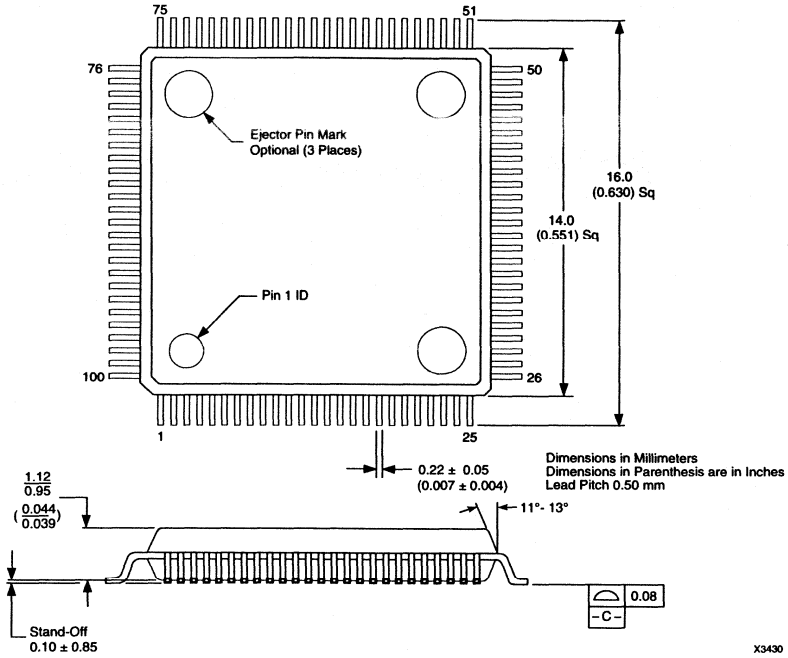
100-Pin Plastic PQFP (PQ100)



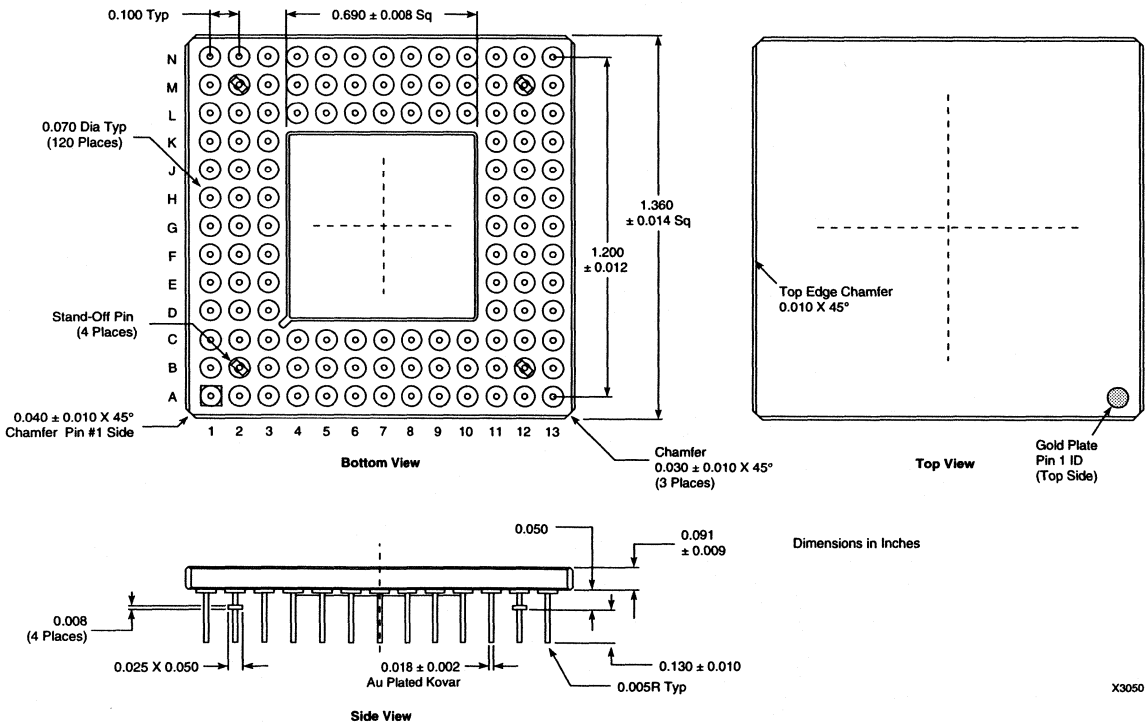
100-Pin Plastic TQFP (TQ100)



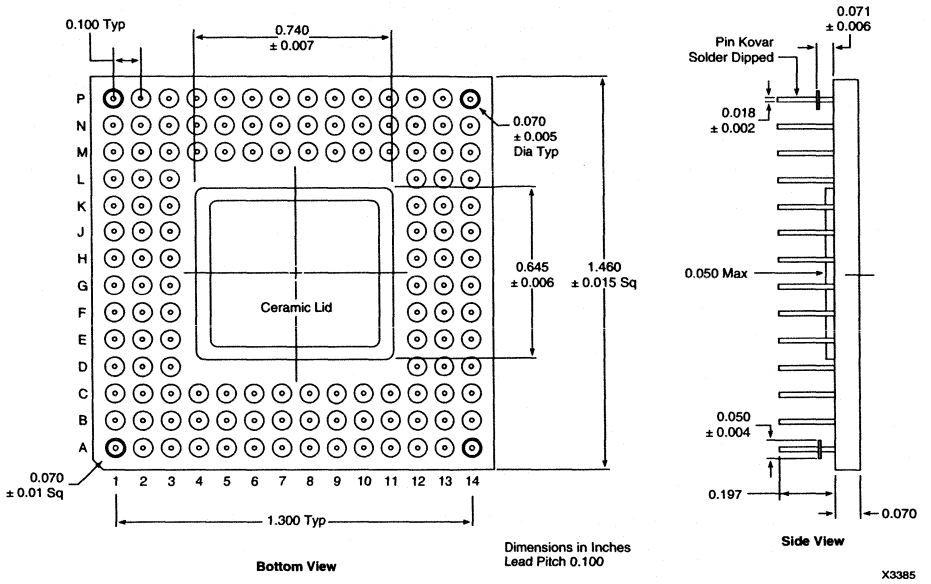
100-Pin Ceramic QFP (CB100)



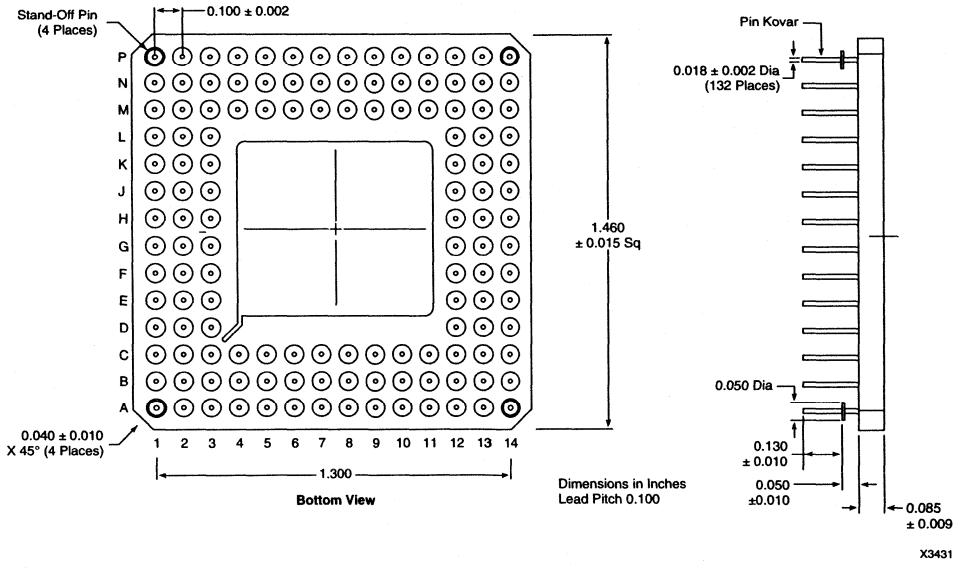
100-Pin Plastic VQFP (VQ100)



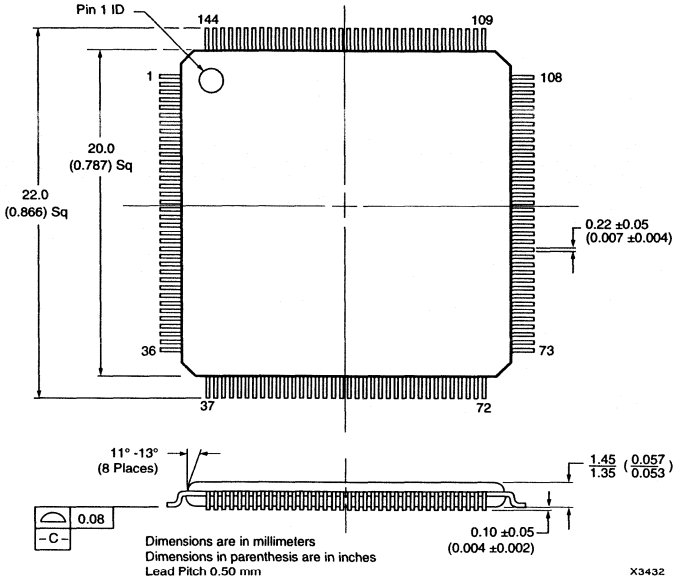
120-Pin Ceramic PGA (PG120)



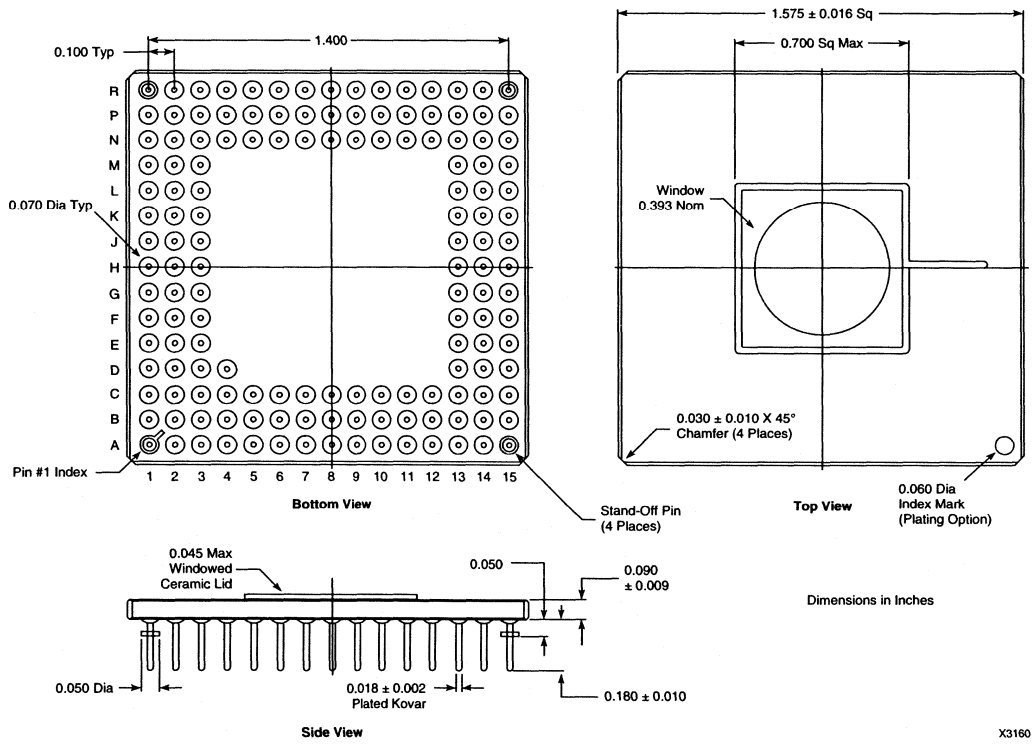
132-Pin Plastic PGA (PP132)



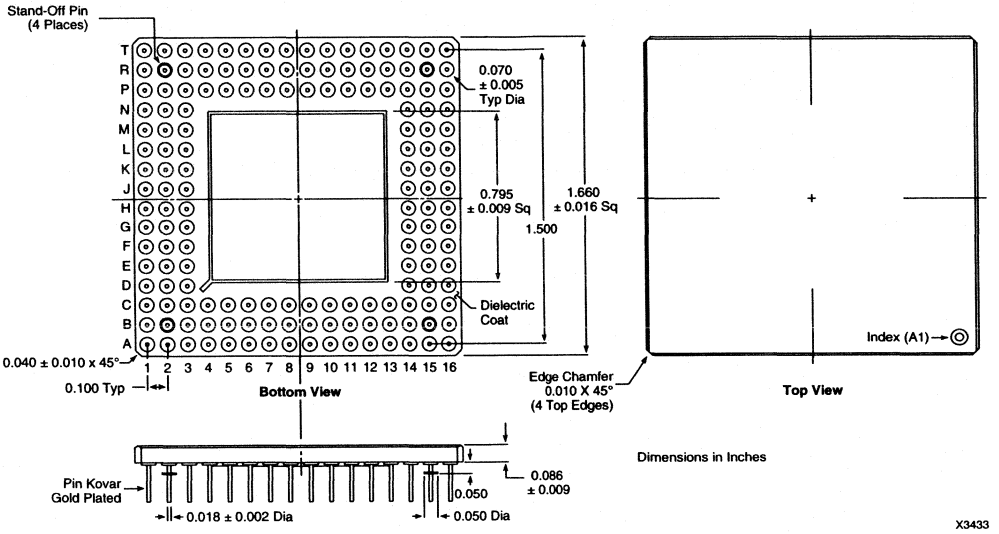
132-Pin Ceramic PGA (PG132)



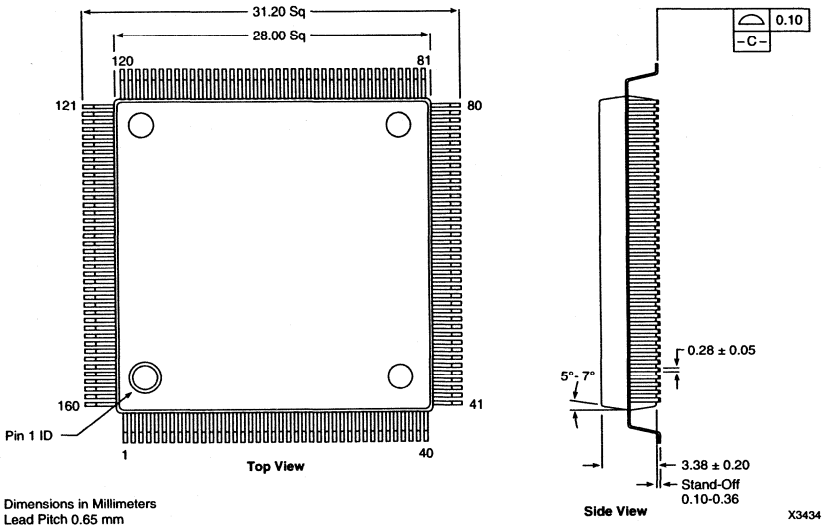
144-Pin Plastic TOFP (TQ144)



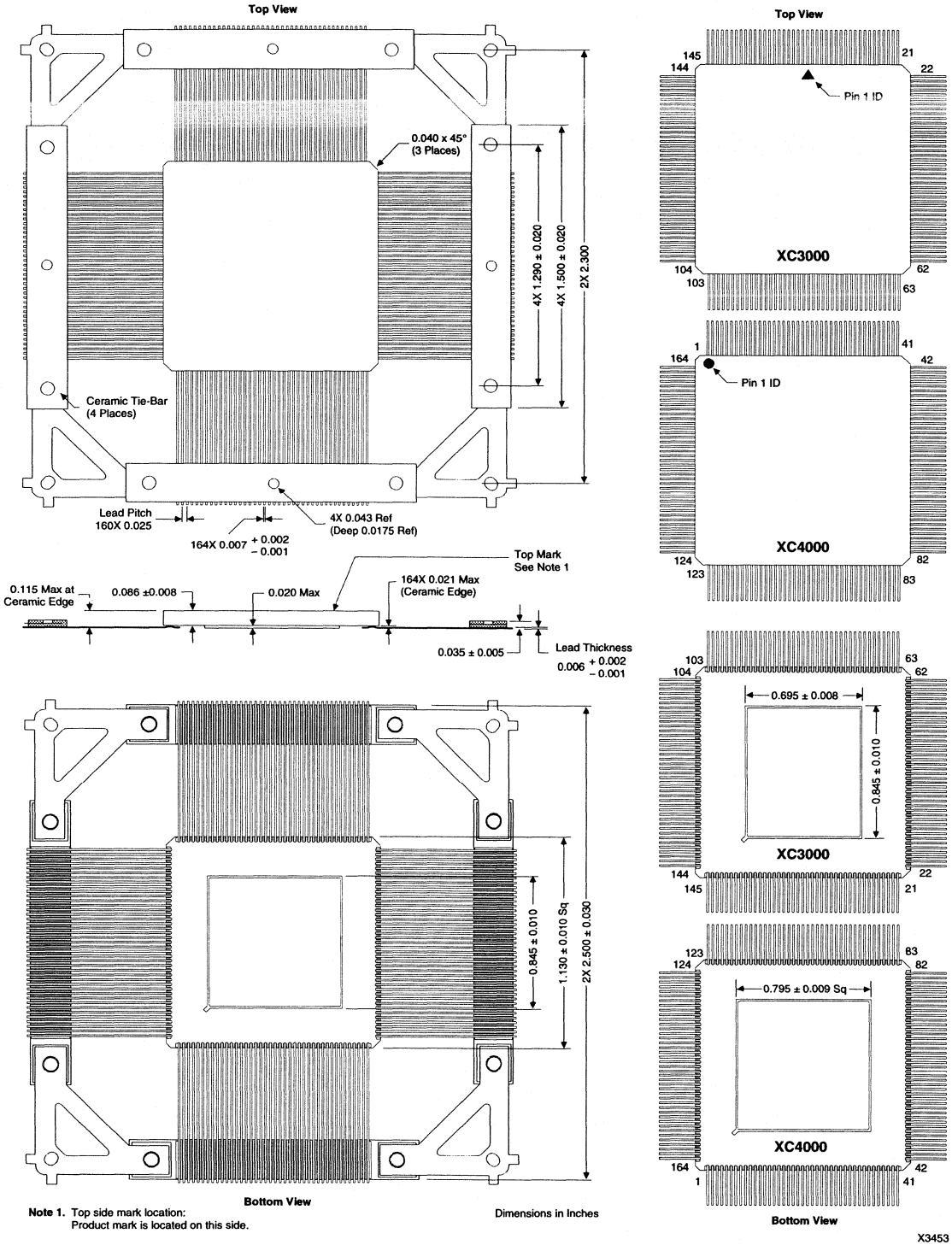
144-Pin Ceramic PGA (PG144)



156-Pin Ceramic PGA (PG156)



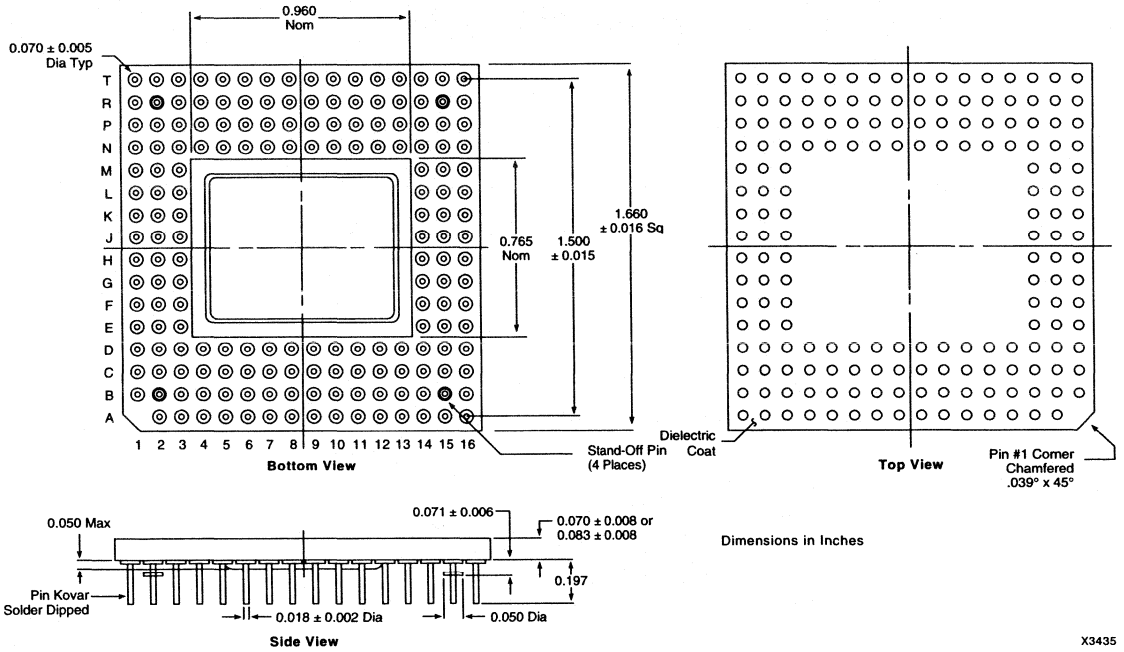
160-Pin Plastic PQFP (PQ160)



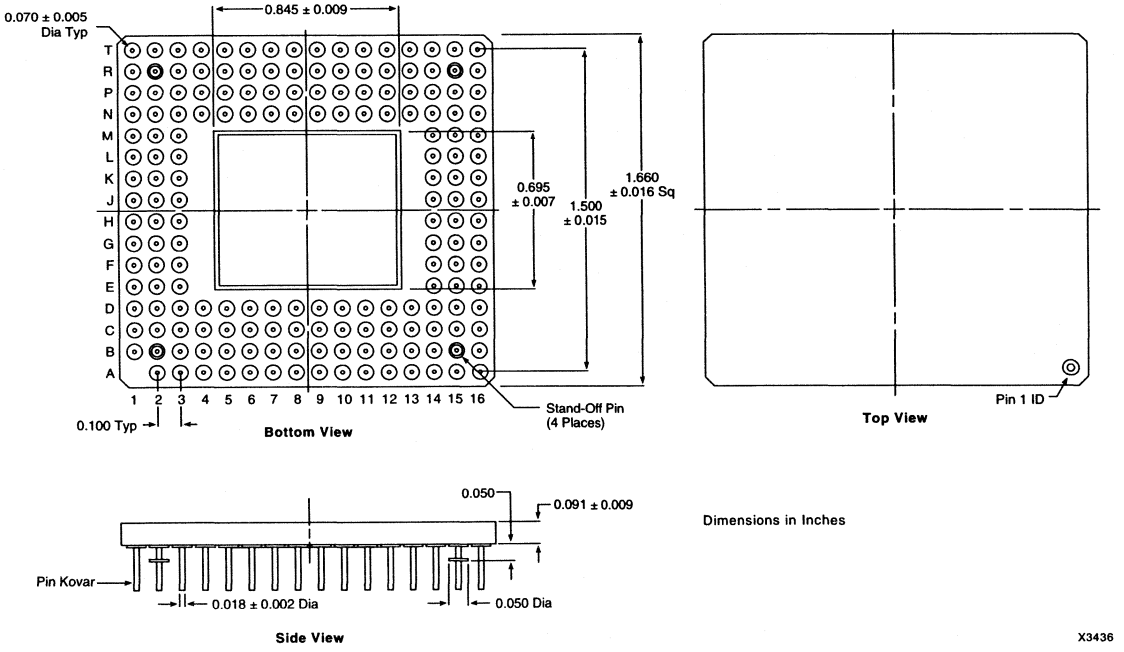
Note 1. Top side mark location:
Product mark is located on this side.

Dimensions in Inches

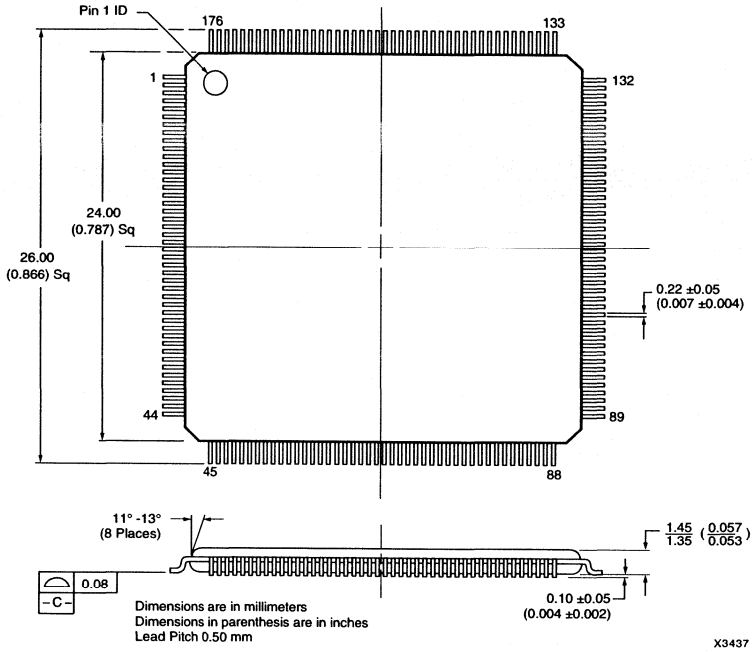
164-Pin Ceramic QFP (CB164)



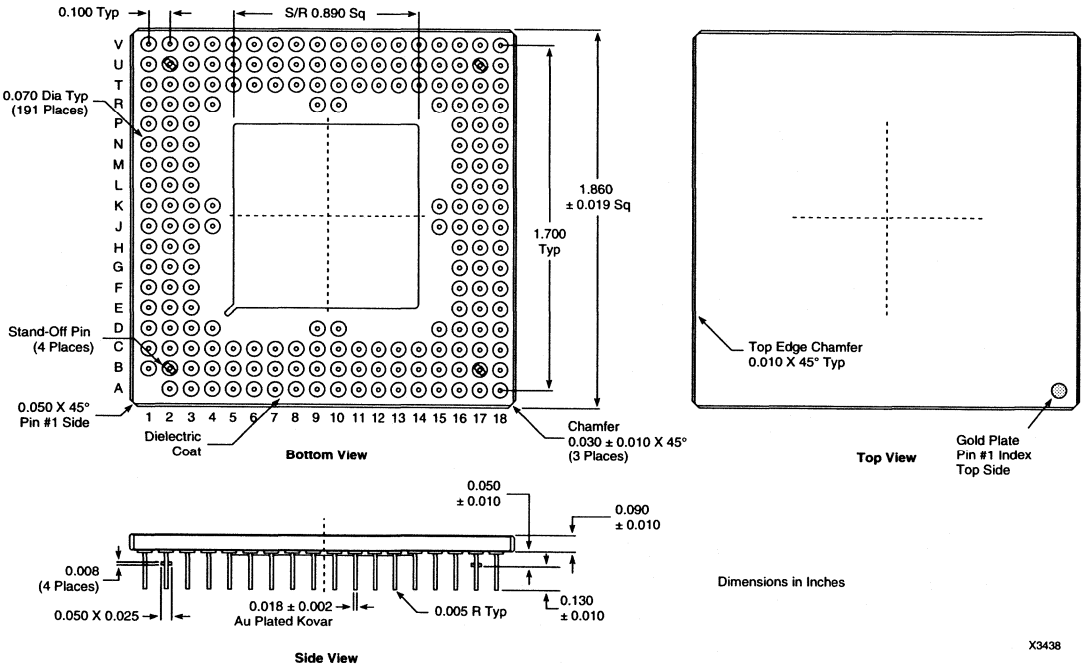
175-Pin Plastic PGA (PP175)



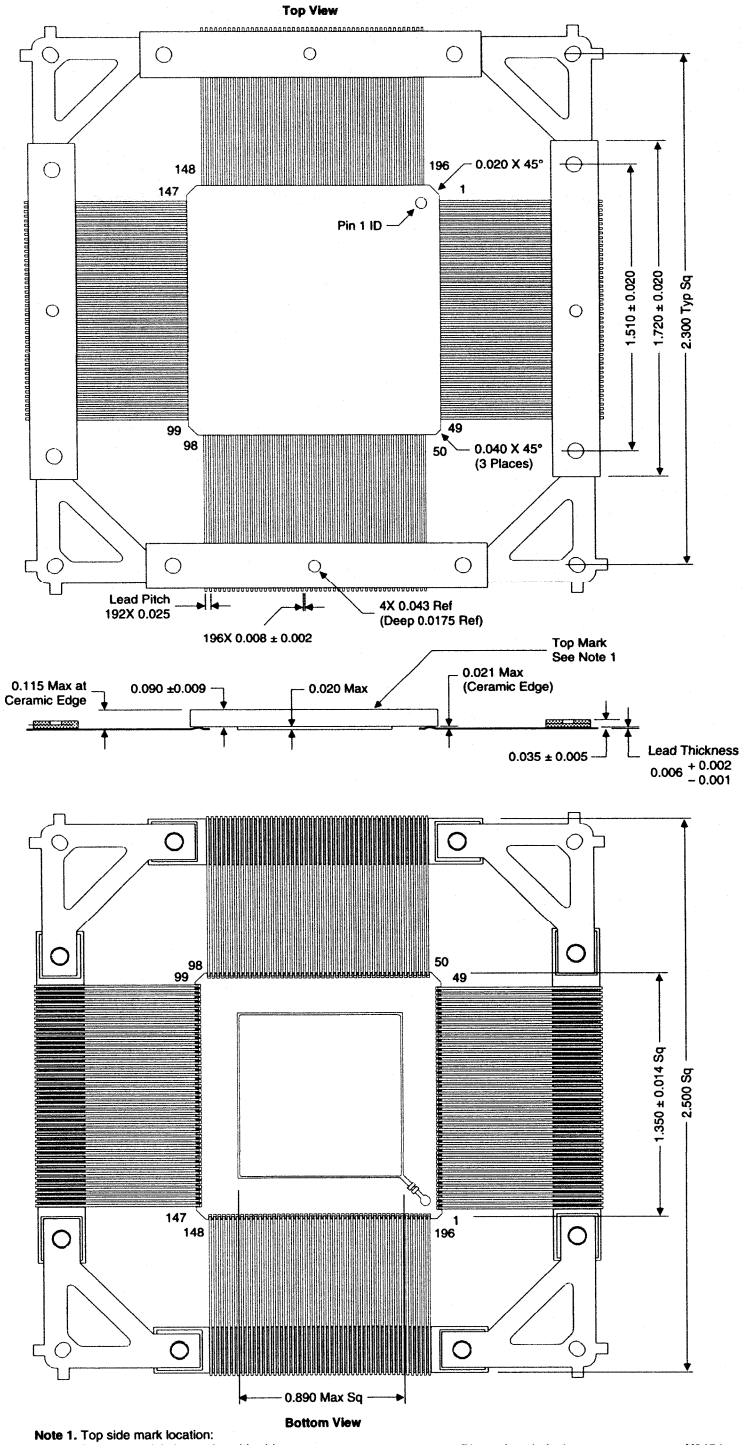
175-Pin Ceramic PGA (PG175)



176-Pin Plastic TOFP (TQ176)



191-Pin Ceramic PGA (PG191)

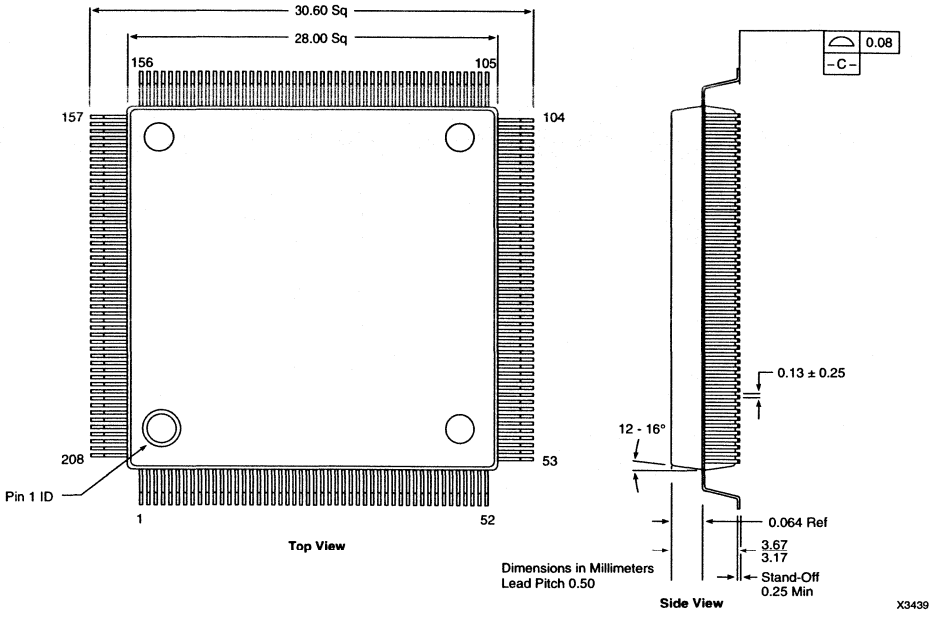


Note 1. Top side mark location:
Product mark is located on this side.

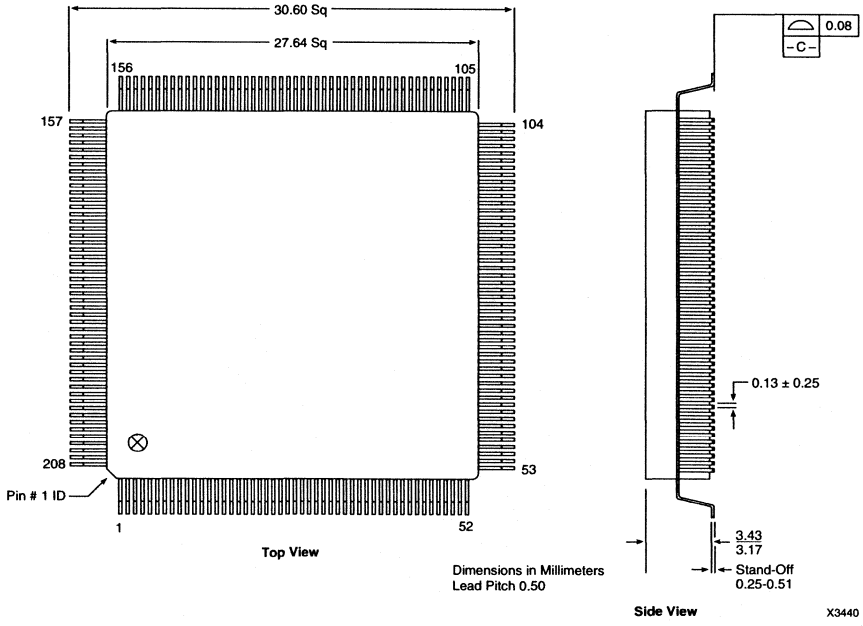
Dimensions in Inches

X3454

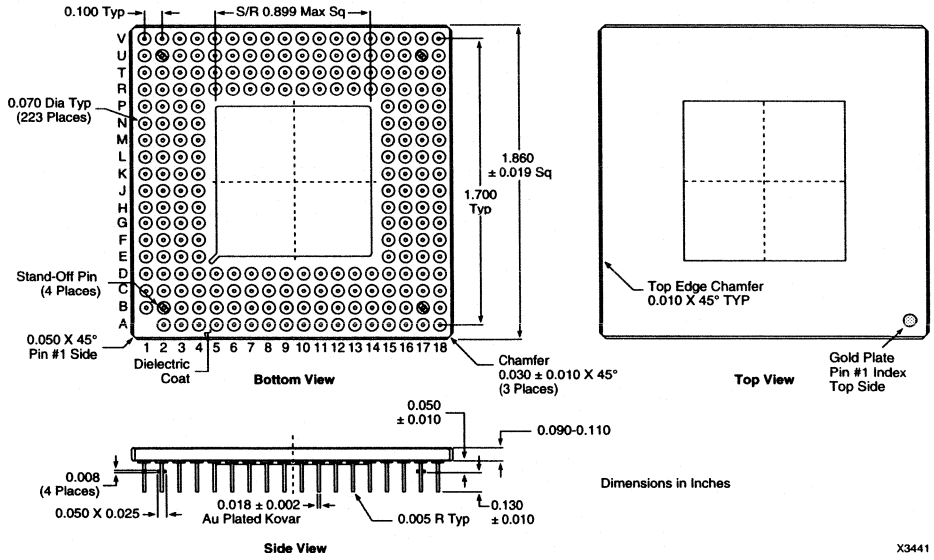
196-Pin Ceramic QFP (CB196)



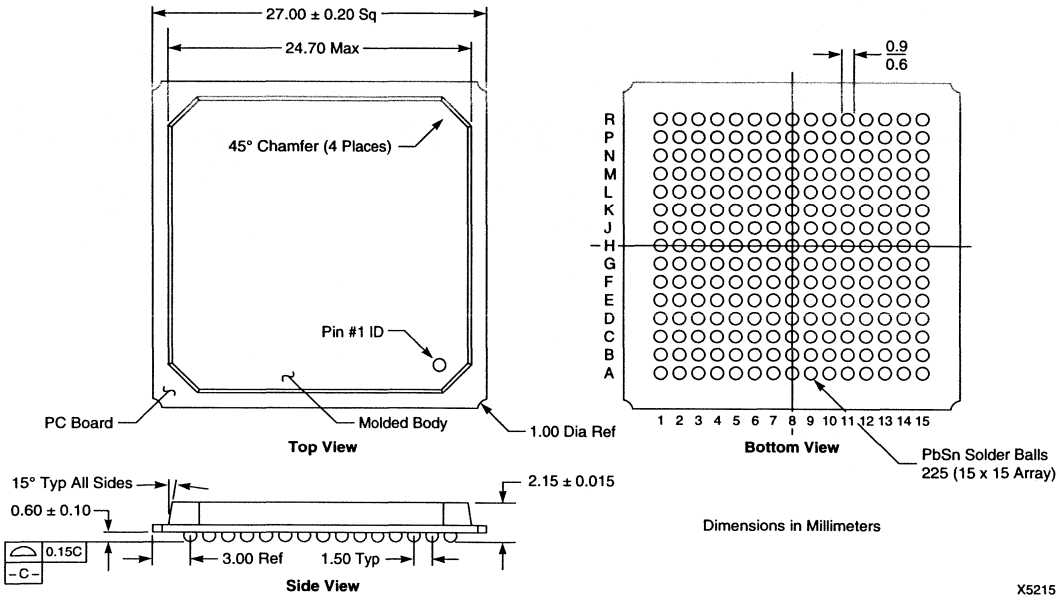
208-Pin Plastic PQFP (PQ208)



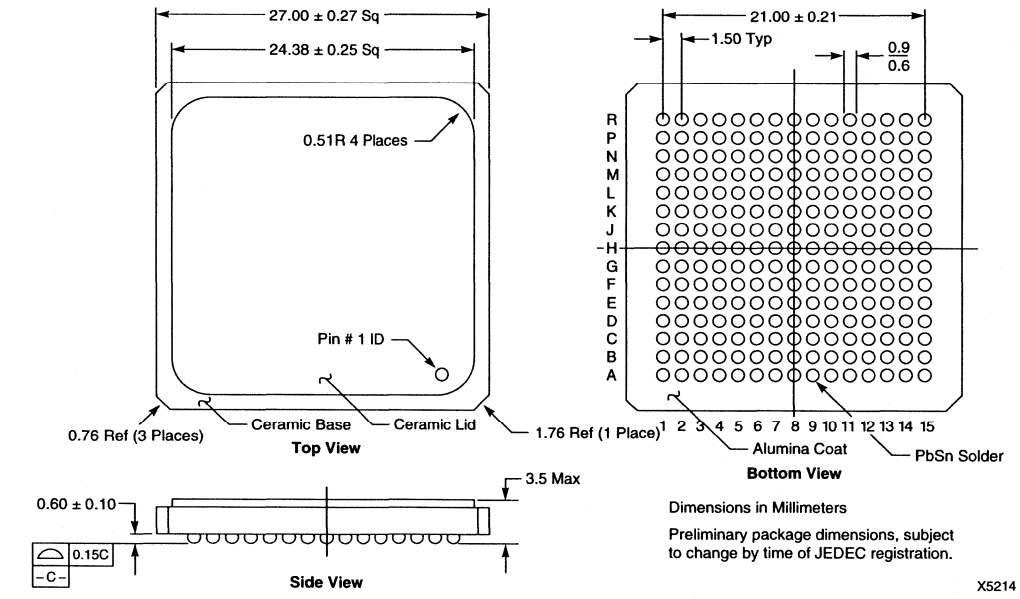
208-Pin Metal MQFP (MQ208)



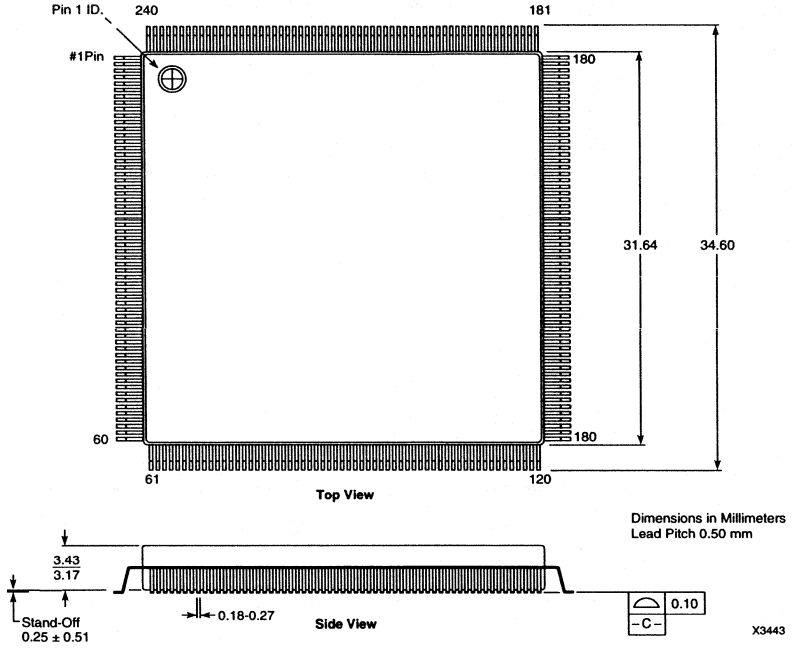
223-Pin Ceramic PGA (PG223)



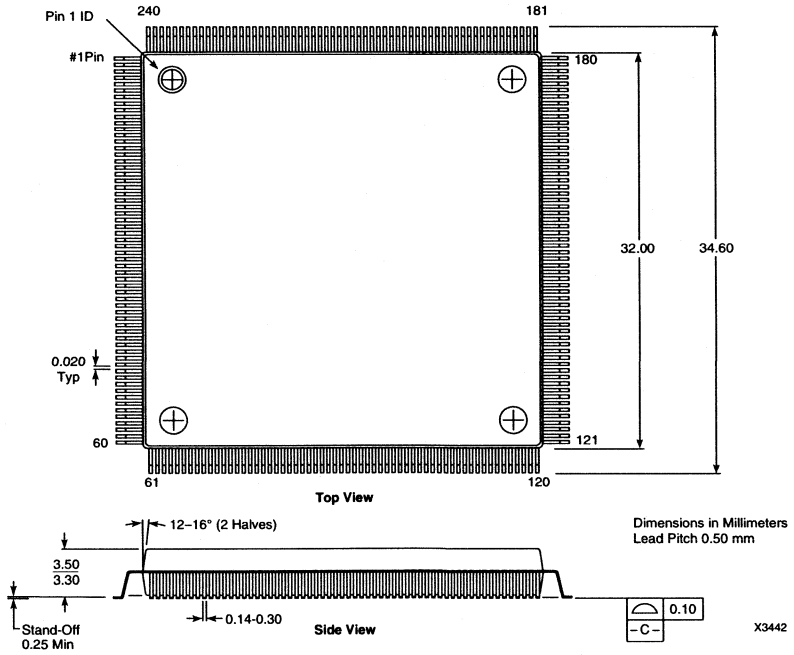
225-Pin Plastic BGA (BG225)



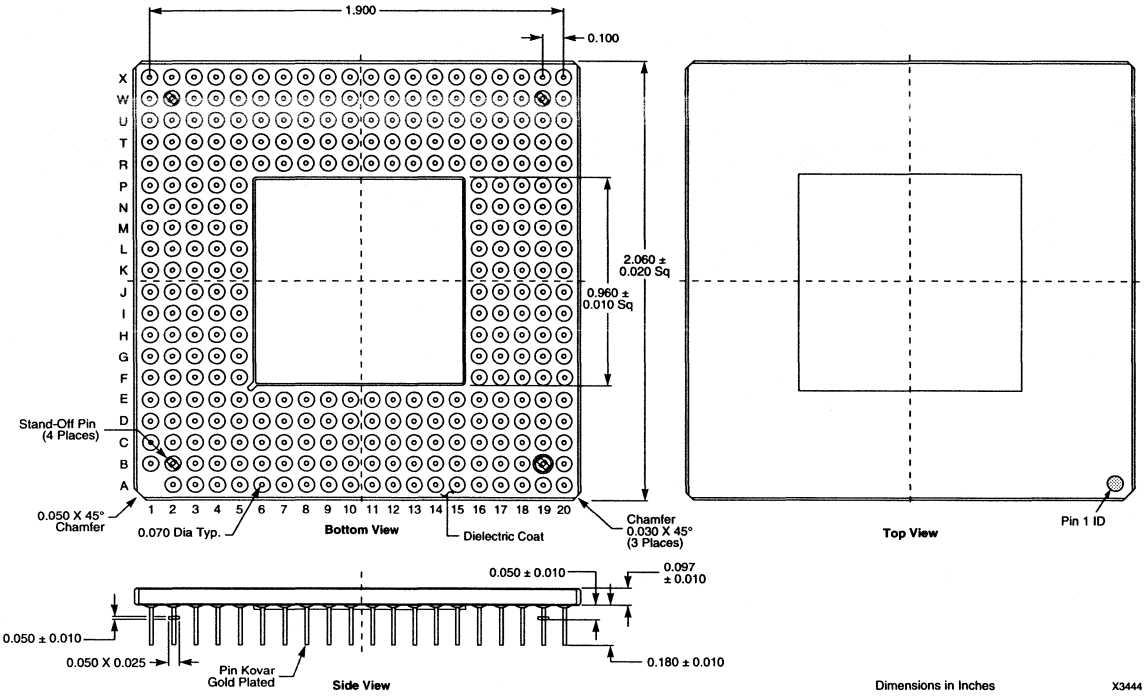
225-Pin Ceramic (CG225)



240-Pin Metal MQFP (MQ240)



240-Pin Plastic PQFP (PQ240)



299-Pin Ceramic PGA (PG299)

Package Thermal Characterization Methods & Conditions

Method and Calibration

Xilinx uses the indirect electrical method for thermal-resistance characterization of packages. The forward-voltage drop of an isolated diode residing on a special test die is calibrated at a constant forcing current of 0.520 mA with respect to temperature over a correlation temperature range of 22°C to 125°C. The calibrated device is then mounted in an appropriate environment, e.g. still air, forced convection, FC-40, etc. Power (Pd) is applied to the device through diffused resistors on the same thermal die; usually between 0.5 to 4 W is applied, depending on the package. The resulting rise in junction temperature (T_J) is monitored with the forward-voltage drop of the pre-calibrated diode. Typically, three identical samples are tested at each data point. The reproducibility error is close to 6%.

Junction-to-Case Measurement – θ_{JC}

The junction-to-case characterization is measured in a 3M Flourinert (FC-40) isothermal circulating fluid stabilized at 25°C. During the measurement, the Device Under Test (DUT) is completely immersed in the fluid; initial stable conditions are recorded, then Pd is applied. Case temperature (T_C) is measured at the primary heat-flow path of the particular package. Junction temperature (T_J) is calculated from the diode forward-voltage drop from the initial condition before power is applied, i.e.

$$\theta_{JC} = \frac{T_J - T_C}{Pd}$$

The junction-to-isothermal-fluid measurement θ_{JL} can also be calculated from the above data as follows:

$$\theta_{JC} = \frac{T_J - T_L}{Pd}$$

where T_L = isothermal fluid temperature.

The latter data is considered as the ideal θ_{JA} data for the package that can be obtained with the most efficient heat removal scheme–airflow, copper-clad board, heat sink or some combination of these. Since this is not a widely used parameter in the industry, and it is not very realistic for normal application of Xilinx packages, the data are not published. The thermal lab keeps such data for package comparisons.

Junction-to-Ambient Measurement – θ_{JA}

θ_{JA} is measured on a 4.5" x 6.0" x .0625" (11.4 cm x 15.2 cm x 0.16 cm) FR-4 board. The data may be taken with the package in a socket or, for packages used primarily for surface mount, with the package mounted directly on traces on the FR-4 board. The copper-trace density is limited to the pads needed for the leads and the 10 or so traces required for signal conditioning and measurement. The board is mounted in a cylindrical enclosure and data is taken at the prevailing temperature and pressure– between 22°C and 25°C ambient (T_A). The power application and signal monitoring proceed in the same way as the θ_{JC} measurement with enclosure (ambient) thermocouple substituted for the fluid thermocouple and two extra thermocouples brought in to monitor room and board temperatures. The junction-to-ambient thermal resistance is calculated as follows:

$$\theta_{JA} = \frac{T_J - T_A}{Pd}$$

The setup lends itself to the application of various airflow velocities from 0 - 800 Linear Feet per Minute (LFM), i.e., 0 - 4.06 m/s. Since the board selection (copper trace density, mounting distance, board thermal conductivity etc) affects the results of the thermal resistance, the data from these tests shall always be qualified with the board-mounting information.

Data Acquisition and Package Thermal Database

Data for a package type is gathered for various die sizes, power levels, cooling modes (airflow and sometimes heat-sink effects) with an IBM-PC based Data Acquisition and Control System (DAS). The system controls and conditions the the power supplies and other ancillary equipment for a hands-free data taking. Different custom-tailored setups within the DAS software are used to run calibration, θ_{JA}, θ_{JC}, fan test as well as power-effects characteristics of a package. A package is completely characterized with respect to the major variables that influence the thermal resistance. A database is generated for the package. From the database, thermal resistance data is interpolated as typical values for the individual Xilinx devices that are assembled in the characterized package. (See data in following tables.)

Thermal Resistance, Junction-to-Ambient (θ_{JA} in °C/W)

Pins	44	48	64	68		84		100				120	132	
Type	Plastic PLCC	Plastic DIP	Plastic VQFP	Plastic PLCC	Ceramic PGA	Plastic PLCC	Ceramic PGA	Plastic PQFP	Plastic TQFP	Plastic VQFP	Top-Brazed CQFP	Ceramic PGA	Plastic PGA	Ceramic PGA
Code	PC44	PD48	VQ64	PC68	PG68	PC84	PG84	PQ100	TQ100	VQ100	CB100	PG120	PP132	PG132
XC2064		43.2		42.1										
XC2018				40.7		35.1	38.4							
XC3020				40.9		35.3	35.4	75.3			45.3			
XC3030	43.6			39.4		33.7	33.7	71.1						
XC3042						32.3	32.3	68.1			44.2		33.7	26.5
XC3064						30.5							32.9	24.1
XC3090						29.1								
XC3195														
XC4002A						32.9		60.3		80.0E				
XC4003														
XC4003A						31.7		56.2		80.0E				
XC4003H														
XC4004A						30.3								
XC4005						28.5								
XC4005A						30.3								
XC4005H														
XC4006														
XC4008														
XC4010														
XC4013														
XC7236	44.1													
XC7272				39.1		33.3								
XC73108						32.2								

XC17000 Family < 125°C/W in any package
Sub-families, like XC3000A, XC3100, XC3000L are not listed since their die sizes and, therefore, thermal resistances are very similar to those of the main families.

X5257

Thermal Resistance, Junction-to-Case (θ_{JC} in °C/W)

Pins	44	48	64	68		84		100				120	132	
Type	Plastic PLCC	Plastic DIP	Plastic VQFP	Plastic PLCC	Ceramic PGA	Plastic PLCC	Ceramic PGA	Plastic PQFP	Plastic TQFP	Plastic VQFP	Top-Brazed CQFP	Ceramic PGA	Plastic PGA	Ceramic PGA
Code	PC44	PD48	VQ64	PC68	PG68	PC84	PG84	PQ100	TQ100	VQ100	CB100	PG120	PP132	PG132
XC2064		11.6		9.5										
XC2018				8.1		6.7	9.2		20.0					
XC3020				8.3		6.9	6.1	9.3						
XC3030	10.7			7.0		5.7	5.4	6.8	20.0					
XC3042						4.8	4.9	5.0	20.0				2.1	2.5
XC3064						3.6							1.9	2.0
XC3090						2.8								
XC3195														
XC4002A						5.2		5.8						
XC4003														
XC4003A						4.4		4.4						
XC4003H														
XC4004A						3.5								
XC4005						2.5								
XC4005A						3.5								
XC4005H														
XC4006														
XC4008														
XC4010														
XC4013														
XC7236	12.2													
XC7272				6.8		5.5	5.1							
XC73108						4.7								

XC1700 Family < 30°C/W in any package
Sub-families, like XC3000A, XC3100, XC3000L are not listed since their die sizes and, therefore, thermal resistances are very similar to those of the main families.

X5256

Thermal Resistance, Junction-to-Ambient (θ_{JA} in °C/W)

Pins	144	156	160	164	175		176	191	196	208		223	240	
Type	Plastic TQFP	Ceramic PGA	Plastic PQFP	Top-Brazed CQFP	Plastic PGA	Ceramic PGA	Plastic TQFP	Ceramic PGA	Top-Brazed CQFP	Plastic PQFP	Metal MQFP	Ceramic PGA	Plastic PQFP	Metal MQFP
Code	TQ144	PG156	PQ160	CB164	PP175	PG175	TQ176	PG191	CB196	PQ208	MQ208	PG223	PQ240	MQ240
XC2064														
XC2018														
XC3020														
XC3030														
XC3042														
XC3064			33.0											
XC3090			31.8	26.3	21.7	16.4				30.5				
XC3195														
XC4002A														
XC4003														
XC4003A														
XC4003H								20.0E		32.0				
XC4004A														
XC4005			30.8	24.0E						31.4				
XC4005A			31.3							31.9				
XC4005H												20.0E	28.0	16.0
XC4006			29.7							30.3				
XC4008								20.0E		28.0	13.0E			
XC4010								20.0E		26.8	13.0E			
XC4013											13.0E	20.0E	28.0	16.0
XC7236														
XC7272														
XC73108														

E = Estimated
Sub-families, like XC3000A, XC3100, XC3000L are not listed since their die sizes and, therefore, thermal resistances are very similar to those of the main families.

X5254

Thermal Resistance, Junction-to-Case (θ_{JC} in °C/W)

Pins	144	156	160	164	175		176	191	196	208		223	240	
Type	Plastic TQFP	Ceramic PGA	Plastic PQFP	Top-Brazed CQFP	Plastic PGA	Ceramic PGA	Plastic TQFP	Ceramic PGA	Top-Brazed CQFP	Plastic PQFP	Metal MQFP	Ceramic PGA	Plastic PQFP	Metal MQFP
Code	TQ144	PG156	PQ160	CB164	PP175	PG175	TQ176	PG191	CB196	PQ208	MQ208	PG223	PQ240	MQ240
XC2064														
XC2018														
XC3020														
XC3030														
XC3042														
XC3064			4.4											
XC3090			3.0	1.8	3.0	2.1				2.6				
XC3195												1.8		
XC4002A														
XC4003														
XC4003A														
XC4003H								1.8		4.1				
XC4004A														
XC4005			3.7	1.8						3.5				
XC4005A			4.0							4.1				
XC4005H												1.8	6.0E	2.0E
XC4006			3.0							2.0E				
XC4008								1.8		2.0E	2.0E			
XC4010								1.8		2.0E	2.0E			
XC4013												1.8	6.0E	2.0E
XC7236														
XC7272														
XC73108			5.9											

E = Estimated
Sub-families, like XC3000A, XC3100, XC3000L are not listed since their die sizes and, therefore, thermal resistances are very similar to those of the main families.

X5255

Component Mass (Weight) by Package Type

Package	Description	Mass (grams)
CB100-1	NCTB - Top Brazed Ceramic -4K	10.80
CB100-2	NCTB - Top Brazed Ceramic -3K	10.50
CB164-1	NCTB - Top Brazed Ceramic -3K	11.20
CB164-2	NCTB - Top Brazed Ceramic -4K	11.50
CB196	NCTB - Top Brazed Ceramic	15.30
CQ100	0.025" Unformed CERQuad	3.60
CQ164	0.025" Unformed CERQuad	8.35
DD8	0.300 CERDip	1.07
MQ208	Metal Quad (EIAJ 28 mm)	6.10
MQ240	32 X 32 mm	7.80
PC20	PLCC - JEDEC 0.050"	0.75
PC44	PLCC - JEDEC 0.050"	1.20
PC68	PLCC - JEDEC 0.050"	4.80
PC84	PLCC - JEDEC 0.050"	6.80
PD48	Dual In Line Plastic - 0.600"	7.90
PD8	Dual In Line Plastic - 0.300"	0.52
PG120	PGA 13 X 13 Matrix Ceramic	11.50
PG132	PGA 14 X 14 Matrix Ceramic	11.75
PG156	PGA 16 X 16 Matrix Ceramic	17.10
PG175	Heat Sink - 16X16 Matrix KCW10	28.40
PG175	No Heat Sink - 16X16 Matrix	17.70
PG191	PGA 18 X 18 Matrix Ceramic	21.80
PG223	PGA 18 X 18 Matrix Ceramic	26.00
PG68	Cav. Up CPGA 11 X 11 Matrix	6.95
PG84	Cav. Up CPGA 11 X 11 Matrix	7.25
PP132	Plastic PGA 14 X 14 Matrix	8.10
PP156	PPGA 16 X 16 Matrix	10.60
PP175	16 X 16 PPGA 2 Tier - Hardware Ver.	10.00
PP175	16 X 16 PPGA Exposed Copper Ver.	9.90
PP175	16 X 16 PPGA Buried Copper Ver.	11.08
PQ100	EIAJ - Matrix (14X20)	1.60
PQ160	EIAJ - Matrix 1.6 mm Form	5.80
PQ208	EIAJ - 28 mm BODY 1.3 mm Form	5.25
PQ240	32 X 32 mm	7.05
PQ256	EIAJ - 40 X 28 Metric	–
SO8	SOIC Narrow 0.150 Body	0.08
TQ100	Thin QFP 1.4 mm thick	0.65
TQ144	20 X 20 mm 1.4 mm thick	1.35
TQ176	24 X 24 mm 1.4 mm thick	1.75
VQ64	10 X 10 mm 1.0 mm thick	0.60
VQ100	14 X 14 mm 1.0 mm thick	0.63
WC44	Windowed CERQuad - JEDEC	2.85
WC68	Windowed CERQuad - JEDEC	7.35
WC84	Windowed CERQuad - JEDEC	10.95
WG84	Windowed PGA 11 X 11 Matrix	10.80
WG144	Windowed PGA 15 X 15 Matrix	16.90

* Data represents average values for typical packages with typical devices. For accuracy between 7% to 10%, these numbers will be adequate.

* More precise numbers (below 5% accuracy) for specific devices may be obtained from Xilinx through a factory representative.

Surface-Mount Precautions for PLCCs and PQFPs

Moisture-Induced Cracking During Solder Reflow

The reflow-soldering processes employed in attaching some plastic surface mount components (PSMC) to circuit boards expose the components to very high temperatures and steep temperature gradients. If the component has absorbed sufficient moisture, the plastic overmold may crack. The moisture trapped in the encapsulant vaporizes during the reflow-soldering operation and generates hydrostatic pressure within the package.The pressure may be sufficient in some package-die combinations to cause delamination within the package, or worse, an internal or external crack in the overmold. Cracks in the overmold allow flux and other contaminants to reach the die area and subsequently lead to the early failure of these cracked PSMCs.

Xilinx reliability tests, which include moisture precondition to 0.12% by package mass, have shown no failures attributable to the type of failure described herein. However, the cracking conditions have been duplicated in some package-die combinations under special moisture-saturation conditions. The conditions were part of a general crack-susceptibility characterization to determine what packages, if any, were likely to experience the failure. Current findings, confirmed by industry studies, show that the 20PLCC, 44PLCC and 68PLCC exhibit minimal to no tendency to moisture-induced cracking. Other packages have different moisture thresholds for cracking. The important conclusion is that below 0.12% by mass of moisture – corresponding to 168 hours of 30%RH at 85°C – none of the Xilinx packages crack.

In view of these findings from the susceptibility studies, it is necessary to issue special handling precautions for PSMCs, to be applied prior to reflow soldering operation. The crack susceptibility of PSMC is affected by several variables. Among them are the package construction detail – material, design, geometry, die size, package thickness, assembly, etc.–, moisture absorbed, the reflow soldering conditions, etc. One controllable factor is the level of moisture absorbed by the package prior to reflow. Xilinx recommends, in line with industry practice, that all PLCCs, with lead counts above or equal to 44, and all Plastic Quad Flat Packs (PQFPs) be used dry in surface-mount applications. The recommendation is not applicable to PSMCs intended for use in socket applications. For the purpose of this note, a package is considered dry if it has undergone one of the baking schedules listed below, and has been stored at or below 20% RH before reflow operation.

Bake schedules:

a. 24 hours at 125 ±5°C, or

b. 16 hours at 150 ±5°C.

Xilinx Recommendation and Dry Bag Policy

In line with the above recommendation, Xilinx performs dry bake and dry packing on all PQFP shipments. PLCC devices can be done on as needed basis. Contact your Xilinx representative for lead-times, any applicable minimum-order quantities, and pricing. Crack-susceptible PSMCs that ship out of Xilinx without dry bake carry a CAUTION statement on the primary shipping form similar to the Caution Label shown below. Xilinx recommends that PSMC devices that are not dry baked at Xilinx and are intended for surface mount be dry baked prior to reflow, per the instructions on the Caution Label.

Xilinx Dry-Packing Capability

The Xilinx dry-packing program for PQFPs consists of baking the parts after all electrical testing at 125°C for 24 hours in bakeable trays. For PLCC units, the baking is done under similar conditions in aluminum tubes, then transferred to regular shipping forms –tubes or tape. Baked units in shipping forms are sealed within 24 hours under controlled environment in special Moisture Barrier Bags (MBBs).

Enough desiccant pouches are enclosed in the bags to maintain the content at less than 20% RH for up to12 months from the date of seal. A reversible humidity indicator card (HIC) is enclosed to monitor the internal humidity level. The loaded bag is then sealed shut under partial vacuum with an impulse heat sealer. Finally labels are attached to the MBB to alert the customers of the need for special handling precautions. Besides the application information found on the bags, the following handling precautions shall be noted.

Handling of Parts in Sealed Bags

Inspection

Note the seal date and verify that the bag has no holes, tears or punctures that may expose the contents. Review the content information against the parts ordered. It is recommended that the bag remain closed until the contents are ready for use.

Storage

The sealed MMB should be stored unopened in a relatively dry environment of no more than 90% relative humidity and 40°C. The enclosed HIC is the only verification to show if the parts have been exposed to moisture or not.

Expiration Date

The seal date is stated on the bag. The expiration date is 12 months from the seal date. If the expiration date has been exceeded, or if upon opening a bag within its stated expiration period, the HIC shows humidity over 30%, proceed as follows. Bake the components per the bake schedules stated earlier. After baking, any of the following options may apply.

Use the parts within 48 hours

Reseal the parts in a MBB within 12 hours after baking with fresh desiccant pouches and HIC;

Store the baked parts in a controlled cabinet with less than 20% RH. A desiccator cabinet with controlled RH would be ideal.

Other Conditions

Open the MBB when parts are ready to be used. The bag may be opened by cutting across the top as close to the seal as possible. This gives room for possible reseal. When the bag is opened, follow the guidelines under the factory-floor-life section to ensure that devices are maintained below the critical moisture levels. Bags opened for less than an hour (strongly dependent on environment) may be resealed with the original desiccant and HIC. If the bag is not resealed immediately, new desiccant or the old one that has been dried out may be used to reseal, provided that the factory-floor life has not been exceeded. Note that the factory-floor life is cumulative. The clay-based desiccant pouches used by Xilinx may be dried out at 120°C ±5°C for 10 to 16 hours. Fresh desiccants may be purchased from United Desiccant-Gates, USA (Model: Desi Pack C, 2 unit desiccant in Tyvek bag). Note also that the Humidity Indicator Card is reversible and may be reused.

Factory Floor Life

The maximum life that dry parts may be safely exposed in a manufacturing ambient condition depends on the specpackage that causes moisture-induced cracks.

Some guidelines have been provided by the Institute for Publication IPC-SM-789. Xilinx characterization confirms that if the relative humidity in a factory is kept below 60% with temperature between 25 and 30°C, the parts taken from the MBB will have acceptable moisture levels if used within 48 hours. It is recommended that devices be dry baked if this floor life is exceeded. The time may be extended by use of controlled desiccator cabinet for storage on the floor.

Obviously, Xilinx devices in various Plastic Surface Mount packages are not affected in the same way. As stated earlier some PLCC packages are hardly affected by cracking even under maximum moisture-saturation conditions. In spite of this, the Xilinx current floor-life recommendation is for all PSMCs and is based on data from reliability results on packages with predetermined moisture levels of 0.12%. In general, irrespective of factory floor conditions, Xilinx recommends that devices be dried out if the level of moisture in the package exceeds 0.12% by mass of package. If factory floor conditions are expected to exceed the 30°C/60% RH, please consult Xilinx for more information.

CAUTION

THESE DEVICES REQUIRE BAKING

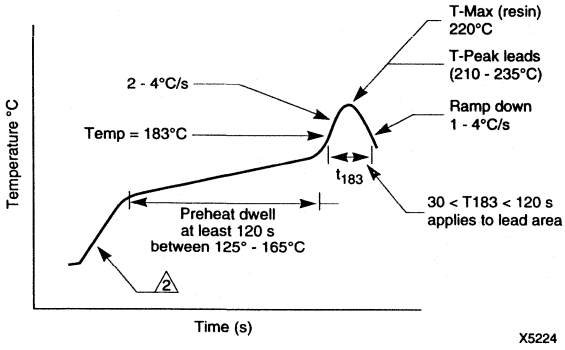
THE ENCLOSED COMPONENTS ARE SENSITIVE TO MOISTURE AND ARE SUSCEPTIBLE TO PACKAGE CRACKING, BOND WIRE BREAKAGE, AND BOND SEPARATION FROM CHIP IF THEY ARE NOT BAKED PRIOR TO ANY EXPOSURE TO HIGH TEMPERATURES OF VAPOR PHASE OR IR REFLOW SOLDERING OR IMMERSION WAVE SOLDERING.

COMPONENT BAKING SHALL BE DONE AT 125°C FOR 24 HRS PRIOR TO ANY REFLOW SOLDERING. UNITS IN TAPE AND REEL AS WELL AS THOSE IN PLASTIC TUBES SHOULD NOT BE SUBJECTED TO THE 125°C BAKE, INSTEAD A LOW TEMPERATURE BAKE (45°C UNTIL 0.11% MOISTURE BY BODY MASS) WILL SUFFICE.

XILINX CAN PROVIDE BAKING AND DRY PACKING SERVICES UPON SPECIAL ORDER.

Plastic Surface Mount Components

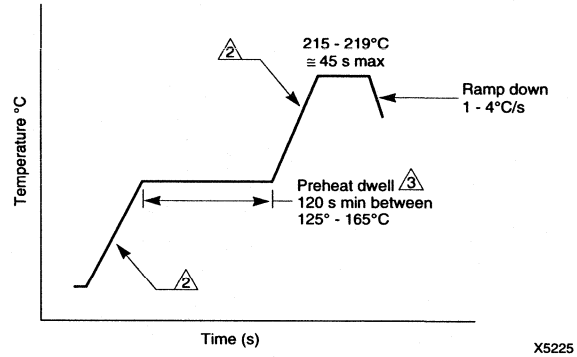
Typical IR reflow profile for board mount is dipicted in Figure 1. The equivalent profile for vapor reflow is shown in Figure 2. The key consideration is that for Plastic Surface Mount Components (PSMC – classified as moisture sensitive plastic components), the package body temperature should not exceed 220° C. In particular, such packages should not be wave soldered or immersed in solder. The maximum heating rates listed should also be observed for the processes. Above all, the components must be used in the *dry condition*. Unless specifically stated for any particular package type, *dry condition* implies less than 0.12% of moisture by package weight. Refer to page 4-29/30 for handling components that are sensitive to moisture.



Notes:
1. Max Resin temperature – not to exceed 220°C, Time at Temp. < 30 s
2. Preheat transition rate < 6°C/s
3. Preheat dwell 125 - 165°C for at least 120 s
4. Reflow transition 4°C/s max
5. IR reflow shall be performed on *dry packages*

The IR process is strongly dependent on equipment and loading differences. Components may overheat due to lack of thermal constraints. Unbalanced loading may lead to significant temperature variation on the board. This guideline is intended to assist users to avoid damaging the components. Actual profile should be determined by the users using these guidelines.

Figure 1. Typical Conditions for IR Reflow Soldering



Notes:
1. Solvent - FC5312 – ensures maximum resin temperature of 220°C
2. Transition rate 4 - 5°C/s
3. Dwell is intended for partial dryout
4. These guidelines are for reference. They are based on laboratory runs using *dry packages*. It is recommended that actual packages with known loads be checked with the commercial equipment prior to engaging in mass production.

Figure 2. Typical Conditions for Vapor Phase Reflow Soldering

Sockets

Below are two lists of manufactures known to offer sockets for Xilinx package types. This list does not imply an endorsement by Xilinx. Each user must evaluate the particular socket type.

There are no wire-wrap sockets for PLCCs. One solution is to piggy-back a through-hole PLCC socket mounted in a compatible PGA socket with wire-wrap pins. Note that the board-layout then differs from a PGA board layout.

Zero Insertion Force (ZIF) sockets, recommended for prototyping with 132 and 175 pin PGA devices, also lack the wire-wrap option. Piggy-back the ZIF socket in a normal PGA wire-wrap socket.

PLCC Sockets

AMP Inc.
Harrisburg, PA 17105
(717) 564-0100

Burndy Corp.
Richards Ave.
Norwalk, CT 06856
(203) 852-8437

Garry Electronics
9 Queen Anne Court
Langhorne, PA 19047-1803
(215) 949-2300

Honda - MHOtronics
Deerfield. IL 60015
444 Lake Cook Road, Suite 8
(312) 948-5600

ITT Cannon
10550 Talbert Ave.
P.O.Box 8040
Fountain Valley, CA 92728
(714) 964-7400

Maxconn Inc.
1855 O'Toole Ave., D102
San Jose, CA 95131
(408) 435-8666

Methode Electronics Inc.
1700 Hicks Road
Rolling Meadows, IL 47150
(312) 392-3500

Mill-Max Mfg. Corp.
190 Pine Hollow Road
Oyster Bay, N.Y. 11771-0300
(516) 922-6000

Precicontact Inc.
835 Wheeler Way
Langhorne, PA 19047
(215) 757-1202

Samtec Inc.
P.O.Box 1147
New Albany, IN 47150
(812) 944-6733

3M
Austin, TX
(800) 328-0411

Thomas & Betts Corp.
920 Route 202
Raritan, NJ 08869
(201) 469-4000

Wells Electronics, Inc.
1701 South Main Street
South Bend, IN 46613
(219) 287-5941

Yamaichi - Electronics, Inc.
1420 Koll Circle
Suite B
San Jose, CA 95112
(408) 452-0797

PGA Sockets

Advanced Interconnections
5 Energy Way
West Warwick, RI 02893
(401) 823-5200

AMP Inc.
Harrisburg, PA 17105
(717) 564-0100

Aries Electronics, Inc.
P.O.Box 130
Frenchtown, NJ 08825
(201) 996-6841

Augat
33 Perry Ave.
P.O.Box 779
Attleboro, MA 02703
(617) 222-2202

Bevmar Industries, Inc.
20601 Annalee Ave.
Carson, CA 90746
(213) 631-5152

Bevmar Industries, Inc.
1 John Clarke Rd.
Middletown, RI 02840
(401) 849-4803

Electronic Molding Corp.
96 Mill Street
Woonsocket, RI 02895
(401) 769-3800

Garry Electronics
9 Queen Anne Court
Langhorne, PA 19047-1803
(215) 949-2300

Mark Eyelet Inc.
63 Wakelee Road
Wolcott, CT 06716
(203) 756-8847

McKenzie Technology
44370 Old Warm Springs Blvd.
Fremont CA 94538
(415) 651-2700

Methode Electronics Inc.
1700 Hicks Road
Rolling Meadows, IL 47150
(312) 392-3500

Mill-Max Mfg. Corp.
190 Pine Hollow Road
Oyster Bay, N.Y. 11771-0300
(516) 922-6000

Precicontact Inc.
835 Wheeler Way
Langhorne, PA 19047
(215) 757-1202

Samtec Inc.
P.O.Box 1147
New Albany, IN 47150
(812) 944-6733

Texas Instruments
CSD Marketing, MS 14-1
Attleboro, MA 02703
(617) 699-5206

Thomas & Betts Corp.
920 Route 202
Raritan, NJ 08869
(201) 469-4000

Yamaichi - Electronics, Inc.
1420 Koll Circle
Suite B
San Jose, CA 95112
(408) 452-0797

Quality Assurance and Reliability

Xilinx is committed, as a corporation, to achieving 100% customer satisfaction. This is accomplished by supplying to our customers programmable logic devices and support software that fulfill our commitments for:

on-time delivery of a ***quality*** product that meets customer ***requirements***, and is ***reliable***.

This requires that each employee of Xilinx makes a personal commitment to the attainment of a 6-Sigma level of quality in the fulfillment of all of his or her responsibilities. Each employee is aware of my personal commitment to supply quality parts and products which are fully compliant with all customer requirements.

B. V. Vonderschmitt

Bernard V. Vonderschmitt
President

Quality Assurance Program

All aspects of the Quality Assurance Program at Xilinx have been designed to eliminate the root cause of defects, rather than to try to remove them by inspection. The programs are in full compliance with the requirements of Appendix A of MIL-M-38510. These programs emphasize heavily the aspects of process control, process documentation, and operator training. These programs and those of the company's subcontractors are subjected to extensive internal and external audits to ensure compliance.

Xilinx calculates its outgoing component quality level, expressed in PPM (defective parts per million devices shipped) using the industry standard methods now adopted by JEDEC and published in JEDEC Standard 16. These figures of merit are revised monthly and published quarterly by Xilinx Quality Assurance. Figure 1 shows the quality improvements over the past 18 months.

Xilinx is committed to customer satisfaction. By adhering to the highest quality standards, the company has achieved leadership in the EPLD and FPGA manufacturing areas.

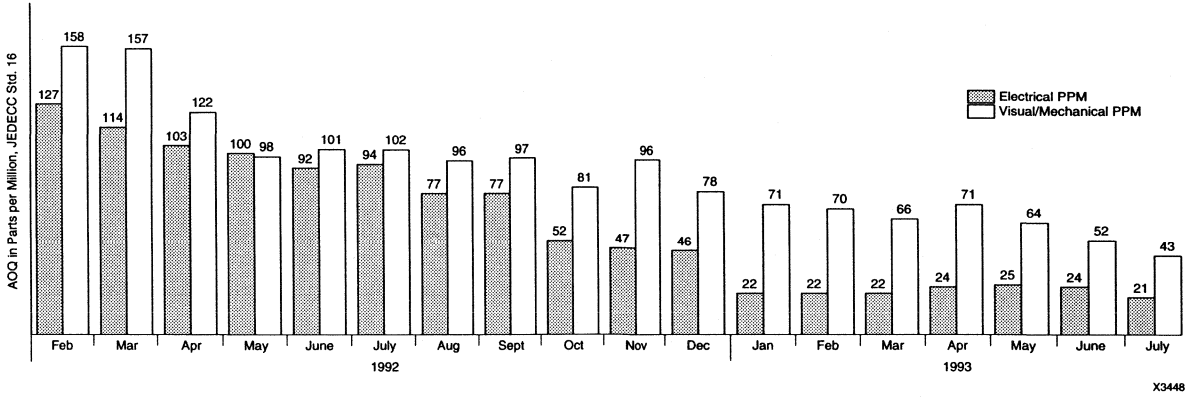


Figure 1. Xilinx Average Outgoing Quality – Mature, High Volume Products

Quality Assurance encompasses all aspects of company business. Xilinx continually strives to improve quality to meet customers' changing needs and expectations. To do this, the company is dedicated to the following.

- To provide a broad range of products and services that satisfy both the expectations of customers and the company's stringent quality standards.
- To emphasize open communications with customers and suppliers, supported with the necessary statistical data.
- To continually improve the quality of Xilinx products, services, and company efficiency.
- To maintain a work environment that fosters quality and reliability leadership and excellence.

Device Reliability

Device reliability is often expressed in a measurement called *Failures in Time* (FITS). In this measure one FIT equals one failure per billion (10^9) device operating hours. A failure rate in FITS must include the operating temperature to be meaningful. Hence failure rates are often expressed in FITS at 70°C (or some other temperature in excess of the application).

Since one billion hours is well in excess of 100,000 years, the FIT rate of modern ICs can only be measured by accelerating the failure rate by testing at a higher junction temperature (usually 125°C or 145°C). Extensive testing of Xilinx XC2000 and XC3000 devices (performed on actual production devices taken directly from finished goods) has been accomplished continuously since 1989. During the last two years, over 7,500 devices (both XC2000 and XC3000) have accumulated a total of over 13,000,000 hours of both static and dynamic operating at 125°C (equivalent) to yield the following FIT rates at 70°C.

	1991	1992				1993		
	Dec	Mar	Jun	Sept	Dec	Mar	Jun	
XC2000, static	8	6	6	4	5	4	4	FITs
XC3000, static	22	20	20	16	15	13	13	FITs
XC3000, dynam	9	9	9	4	2	2	1	FITs

From its inception, Xilinx has been committed to delivering the highest quality, most reliable programmable gate arrays available. A strong Quality Assurance and Reliability program begins at the initial design stages and is carried through to final shipment. An extensive, on-going reliability-testing program is used to predict the field performance of all Xilinx devices.

These tests provide an accelerated method of emulating long-term system operation in severe field environments. From the performance of the devices during these tests, predictions of actual field performance under a variety of conditions can be easily calculated.

This report describes the nature and purpose of the various reliability tests performed on finished devices. Updated summaries are available upon request from the Quality Assurance and Reliability Department at Xilinx.

Outline of Testing

Qualification testing of devices is performed to demonstrate the reliability of the die used in the device, and the materials and methods used in the assembly of the device. Testing methods are derived from and patterned after the methods specified in MIL-STD-883.

Referral to the test methods of MIL-STD-883 is not intended to imply that nonhermetic products comply with the requirements of MIL-STD-883. These test methods are recognized industry-wide as stringent tests of reliability and are commonly used for nonmilitary-grade semiconductor devices, as well as for fully compliant military-grade products.

Hermetic packages are qualified using the test methods specified in MIL-STD-883. The Group D package qualification tests are performed on one lot of each package type from each assembly facility every twelve months.

A summary of the reliability demonstration tests used at Xilinx is contained in Table 1.

Table 1A. Reliability Testing Sequence for Non-Hermetic Devices

Die Qualification

	Name of Test	Test Conditions	Lot Tolerance% Defective Minimum Sample Size/ Maximum Acceptable Failures
1.	High Temperature Life	1000 hr min equivalent at temperature = 125°C Life test circuit equivalent to MIL-STD-883	LTPD = 5
2.	Biased Moisture Life	1000 hr min exposure T = 85°C, RH = 85% Max rated operating voltage Biased moisture life circuit equivalent to MIL-STD-883	LTPD = 5

Non-Hermetic Package Integrity and Assembly Qualification

	Name of Test	Test Conditions	Lot Tolerance % Defective Minimum Sample Size/ Maximum Acceptable Failures
3.	Unbiased Pressure Pot	96 hr min exposure T = 121°C, P = 2 atm H_2O saturated	LTPD = 5
4.	Thermal Shock	MIL-STD-883, Method 1011, Cond. C -65°C to +150°C 200 cycles	LTPD = 5
5.	Temperature Cycling	MIL-STD-883, Method 1010, Cond. C -65°C to +150°C 500 cycles	LTPD = 5
6.	Salt Atmosphere	MIL-STD-883, Method 1009, Cond. A 24 hrs	s = 25, c = 0
7.	Resistance to Solvents	MIL-STD-883, Method 2015	s = 4, c = 0
8.	Solderability	MIL-STD-883, Method 2003	s = 3, c = 0
9.	Lead Fatigue	MIL-STD-883, Method 2004	s = 2, c = 0
10.	Physical Dimensions	MIL-STD-883, Method 2016	s = 15, c = 0

Table 1B. Reliability Testing Sequence for Hermetic Devices

Hermetic Package Integrity and Assembly Qualification

	Name of Test	Test Conditions	Lot Tolerance % Defective Minimum Sample Size/ Maximum Acceptable Failures
1.	Subgroup D1 Physical Dimensions	MIL-STD-883, Method 2016	LTPD = 15
2.	Subgroup D2		LTPD = 15
	a. Lead Integrity	MIL-STD-883, Method 2028	
	b. Seal (fine and gross leak)	MIL-STD-883, Method 1014 (not required for PGAs)	
3.	Subgroup D3		LTPD = 15
	a. Thermal Shock–15 cycles	MIL-STD-883, Method 1011, Cond. B	
	b. Temp. cycling–100 cycles	MIL-STD-883, Method 1010, Cond. C	
	c. Moisture Resistance	MIL-STD-883, Method 1004	
	d. Seal (fine & gross leak)	MIL-STD-883, Method 1014	
	e. Visual Examination	MIL-STD-883, Method 1004 and Method 1010	
	f. End-point electricals	Group A, subgroup 1	
4.	Subgroup D4		LTPD = 15
	a. Mechanical Shock	MIL-STD-883, Method 2002, Cond. B	
	b. Vibration, Variable Freq.	MIL-STD-883, Method 2007, Cond. A	
	c. Constant Acceleration	MIL-STD-883, Method 2001, Cond. E min, Y1 only (Cond. D for large PGAs)	
	d. Seal (fine & gross leak)	MIL-STD-883, Method 1014	
	e. Visual Examination	MIL-STD-883, Method 1010	
	f. End-point electricals	Group A, subgroup 1	
5.	Subgroup D5		LTPD = 15
	a. Salt Atmosphere	MIL-STD-883, Method 1009, Cond. A	
	b. Seal (fine & gross leak)	MIL-STD-883, Method 1014	
	c. Visual Examination	MIL-STD-883, Method 1009	
6.	Subgroup D6 Internal Water Vapor Content	MIL-STD-883, Method 1018, 5000 ppm water at 100°C	s = 3; c = 0 or s = 5; c = 1
7.	Subgroup D7 Lead Finish Adhesion	MIL-STD-883, Method 2025	LTPD = 15, s = 25 leads, (3 device min) c = 0
8.	Subgroup D8 Lid Torque	MIL-STD-883, Method 2024 (for ceramic quad flat pack, CQFP only)	LTPD = 5, s = 5, c = 0

Description of Tests

Die Qualification

1. *High Temperature Life* – This test is performed to evaluate the long-term reliability and life characteristics of the die. It is defined by the Military Standard from which it is derived as a "Die-Related Test" and is contained in the Group C Quality Conformance Tests. Because of the acceleration factor induced by higher temperatures, data representing a large number of equivalent hours at a normal temperature of 25°C can be accumulated in a reasonable period of time.

2. *Biased Moisture Life* – This test is performed to evaluate the reliability of the die under conditions of long-term exposure to severe, high-moisture environments that could cause corrosion. Although it clearly stresses the package as well, this test is typically grouped under the die-related tests. The device is operated at maximum-rated voltage, 5.5 Vdc, and is exposed to a temperature of 85°C and a relative humidity of 85% throughout the test.

Package Integrity and Assembly

Qualification

3. *Unbiased Pressure Pot* – This test is performed at a temperature of 121°C and a pressure of 2 atm of saturated steam to evaluate the ability of the plastic encapsulating material to resist water vapor. Moisture penetrating the package could induce corrosion of the bonding wires and nonglassivated metal areas of the die (bonding pads only for LCA devices). Under extreme conditions, moisture could cause drive-in and corrosion under the glassivation. Although it is difficult to correlate this test to actual field conditions, it provides a well-established method for relative comparison of plastic packaging materials and assembly and molding techniques.

4. *Thermal Shock* – This test is performed to evaluate the resistance of the package to cracking and resistance of the bonding wires and lead frame to separation or damage. It involves nearly instantaneous change in temperature from -65°C to +150°C.

5. *Temperature Cycling* – This test is performed to evaluate the long-term resistance of the package to damage from alternating exposure to temperature extremes. The range of temperatures is -65°C to +150°C. The transition time is longer than that in the Thermal Shock test but the test is conducted for many more cycles.

6. *Salt Atmosphere* – This test was originally designed by the US Navy to evaluate resistance of military-grade shipboard electronics to corrosion from sea water. It is used more generally for non-hermetic industrial and commercial products as a test of corrosion resistance of the package marking and finish.

7. *Resistance to Solvents* – This test is performed to evaluate the integrity of the package marking during exposure to a variety of solvents. This is an especially important test, since an increasing number of board-level assemblies are subjected to severe conditions of automated cleaning before system assembly. This test is performed according to the methods specified by MIL-STD-883.

8. *Solderability* – This test is performed to evaluate the solderability of the leads under conditions of low soldering temperature following exposure to the aging effects of water vapor.

9. *Lead Fatigue* – This test is performed to evaluate the resistance of the completed assembly to vibrations during storage, shipping, and operation.

Testing Facilities

Xilinx has complete capability to perform High Temperature Life Tests, Thermal Shock, Biased Moisture Life Tests, and Unbiased Pressure Pot Tests in its own Reliability Testing Laboratory. Other tests are being performed by outside testing laboratories.

Summary

The testing data in Table 2 shows the actual performance of the devices during the initial qualification tests to which they have been subjected. These test results demonstrate the reliability and expected long life inherent in the non-hermetic product line. This series of tests is ongoing as a part of the Quality Conformance Program on non-hermetic devices.

Table 2. Xilinx Reliability Testing Summary

Device Types: XC17XX, XC2000, XC3000, XC3100, XC4000, XC7200/XC7300
Die Attach Method: Silver Epoxy
Molding Compound: Sumitomo 6300H 7320C

Process/Technology: 1.2, 1.08, 0.8 µ 2-Metal CMOS
Package Type: Varied PLCC/PQ/PPG
Date: 3Q 93

Test	Combined Sample	Failures	Equivalent Mean Hrs/Device at T_A = 125°C	Total Device Hrs at T_A =125°C	Equivalent Failure Rate in FIT at T_J = 70°C
High Temperature Life Test T_A = 145°C (FPGA) T_A = 125°C (EPLD)	18,110	33	3,732	Equivalent Device Hrs 67,604,421	7.2*
Biased Moisture Life Test T = 85°C; RH = 85%	3,852	9	at T_A = 85°C 1,057	at T_A = 85°C 4,073,013	
Unbiased Pressure Pot Test +121°C, 2 atm sat. steam	2,446	0	169	414,168	
Thermal Shock Test -65°C/+150°C 100 cycles (min)	2,587	1	Mean Cycles per Device 300	Total Device Cycles 776,600	
Temperature Cycling Test -65°C/+150°C 200 cycles (min)	3.442	8	Mean Cycles per Device 342	Total Device Cycles 1,177,700	
Salt Atmosphere Test MIL-STD-883, Method 1009, Cond. A	500	0	24	12,000	
Resistance to Solvents Test MIL-STD-883, Method 2105	93	0			
Solderability Test MIL-STD-883, Method 2003	659	1			
Lead Fatigue Test MIL-STD-883, Method 2004	599	0			

* Assumed activated energy 0.90 eV

Data Integrity

Memory Cell Design in the LCA Device

An important aspect of the LCA device reliability is the robustness of the static memory cells used to store the configuration program.

The basic cell is a single-ended 5-transistor memory element (Figure 2). By eliminating a sixth transistor, which would have been used as a pass transistor for the complementary bit line, a higher circuit density is achieved. During normal operation, the outputs of these cells are fixed, since they determine the user configuration. Write and readback times, which have no relation to the device performance during normal operation, will be slower without the extra transistor. In return, the user receives more functionality per unit area.

This explains the basic cell, but how is the LCA user assured of high data integrity in a noisy environment? Consider three different situations: normal operation, a Write operation and a Read operation. In the normal operation, the data in the basic memory element is not changed. Since the two circularly linked inverters that hold the data are physically adjacent, supply transients result in only small relative differences in voltages. Each inverter is truly a complementary pair of transistors. Therefore, whether the output is High or Low, a low-impedance path exists to the supply rail, resulting in extremely high noise immunity. Power supply or ground transients of several volts have no effect on stored data.

The transistor driving the bit line has been carefully designed so that whenever the data to be written is opposite the data stored, it can easily override the output of the feedback inverter. The reliability of the Write operation is guaranteed within the tolerances of the manufacturing process.

In the Read mode, the bit line, which has a significant amount of parasitic capacitance, is precharged to a logic one. The pass transistor is then enabled by driving the

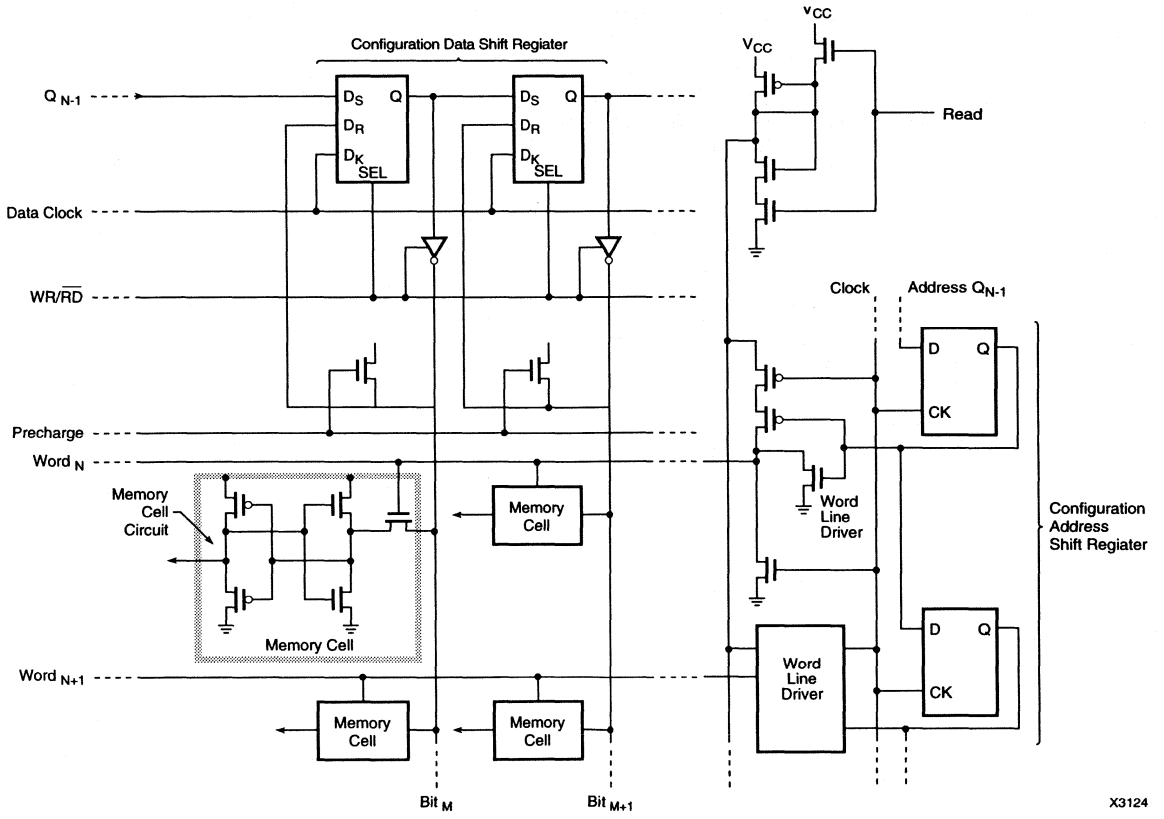


Figure 2. Configuration Memory Cell

word line High. If the stored value is a zero, the line is then discharged to ground. Reliable reading of the memory cell is achieved by reducing the word line High level during reading to a level that insures that the cell will not be disturbed.

Electrostatic Discharge

Electrostatic-discharge (ESD) protection for each pad is provided by circuitry that uses distributed transistors and/or diodes, represented by the circles in Figure 3. In older devices, these protection circuits are conventional diffused structures. In newer designs, Xilinx utilizes proprietary device structures which exhibit substantially enhanced ESD performance (see Table below.)

Whenever the voltage on a pad approaches a dangerous level, current flows through the protective structures to or from a power supply rail (V_{CC} or ground). In addition, the capacitances in these structures integrate the pulse to provide sufficient time for the protection networks to clamp the input, avoiding damage to the circuit being protected. Geometries and doping levels are chosen to provide ESD protection on all pads for both positive and negative voltages.

Circuit Family	Human Body Model 883D Method 3015	Machine Model EIAJ Method 20
XC1700	500 V – 3000 V	250 V – 280 V
XC1700D	8000 V – 9000 V	800 V – 900 V
XC2000	2000 V – 2500 V	250 V – 280 V
XC3000	1000 V – 2900 V	250 V – 425 V
XC3000A	7000 V – 8000 V	600 V – 700 V
XC3100	2500 V – 3500 V	600 V – 700 V
XC4000	7000 V – 9000 V	800 V – 900 V

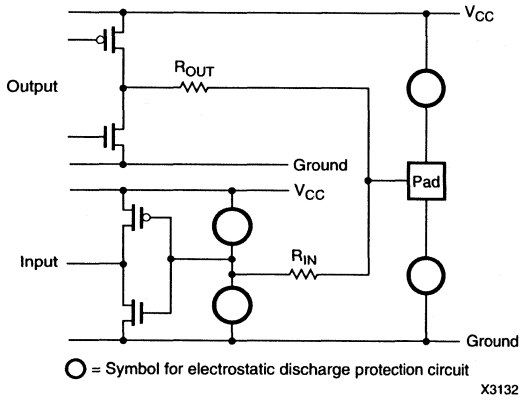


Figure 3. Input/Output Protection Circuitry

Latchup

Latchup is a condition in which parasitic bipolar transistors form a positive feedback loop (Figure 4), which quickly reaches current levels that permanently damage the device. Xilinx uses techniques based on doping levels and circuit placement to avoid this phenomenon. The beta of each parasitic transistor is minimized by increasing the base width. This is achieved with large physical spacings. The butting contacts effectively short the n+ and p+ regions for both wells, which makes the V_{BE} of each parasitic very close to zero. This also makes the parasitic transistors very hard to forward bias. Finally, each well is surrounded by a dummy collector, which forces the V_{CE} of each parasitic almost to zero and creates a structure in which the base width of each parasitic is large, thus making latchup extremely difficult to induce.

At elevated temperatures, 100 mA will not cause latchup. At room temperature, the FPGA can withstand more than 300 mA without latchup; the EPLD device can withstand more than 200 mA without latchup. However, to avoid metal-migration problems, continuous currents in excess of 10 mA are not recommended.

High Temperature Performance

Although Xilinx guarantees parts to perform only within the specifications of the data sheet, extensive high temperature life testing has been been done at 145°C with excellent results.

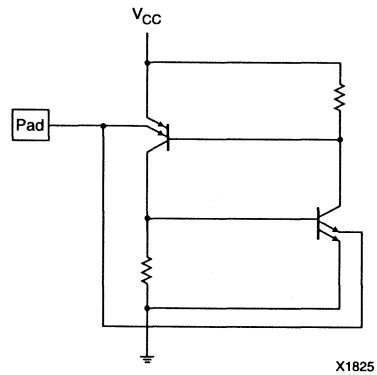
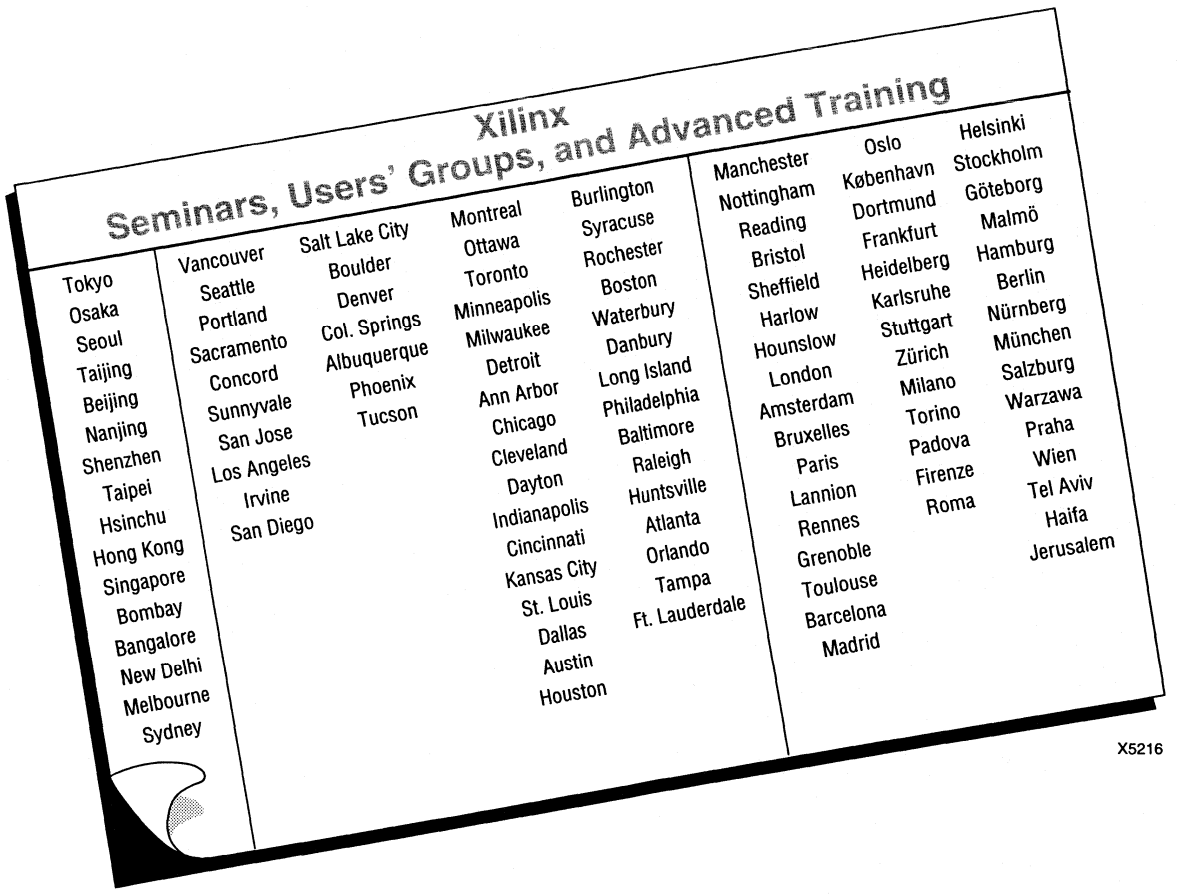


Figure 4. SCR Model

Technical Seminars, Users' Groups, and Advanced Training



Xilinx sponsors technical seminars at locations throughout the world.

Product-oriented seminars are directed toward new and potential users of FPGAs and EPLDs. These seminars include basic descriptions of the Logic Cell Array and EPLD architectures and the benefits of these technologies. Experienced users will also find these seminars useful for learning about newly released products from Xilinx.

Users' Group meetings and Advanced Training sessions are intended for experienced users of Xilinx FPGAs and EPLDs. Emphasis is on taking advantage of the advanced features of the various development system tools to generate FPGA and EPLD designs.

Contact your local Xilinx sales office, sales representative, or distributor for information about seminars in your area.

Xilinx Technical Bulletin Board

FILE COMMANDS

Use these commands to list, download, and upload files.

TYPE:

F <CR> [F)ile Directories]
To list the available File Areas, the files contained in each area, and a short description of each file.

T <CR> [T)ransfer Protocol]
Sets the Transfer Protocol for Uploads/ Downloads. If set to "None", you will be prompted for a Transfer Protocol before each Upload/Download.

D <CR> [D)ownload a File]
To download a file from the Bulletin Board.

U <CR> [U)pload a File]
To upload a file to the Bulletin Board.

FLAG <CR> [FLAG for Download]
To FLAG files for a Batch Download, you must use Ymodem as your Transfer Protocol if you use this option.

DB <CR> [DB Download Batch]
To download multiple files at one time, you must use Ymodem as your Transfer Protocol if you use this option.

UB <CR> [UB Upload Batch]
To upload multiple files at one time, you must use Ymodem as your Transfer Protocol if you use this option.

N <CR> [N)ew Files (date)]
To search for files on the Bulletin Board newer than a specified date.

L <CR> [L)ocate Files (name)]
To search for files on the Bulletin Board by name.

Z <CR> [Z)ippy DIR Scan]
To search for files on the Bulletin Board by text. Will search by both file name and file description.

MESSAGE COMMANDS

Use these commands to send and receive messages.

TYPE:

C <CR> [C)omment to SYSOP]
To leave a message to the System Operator on BBS issues.

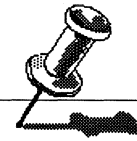
E <CR> [E)nter a message]
To send a message.

R <CR> [R)ead Messages]
To read messages and reply to messages.

K <CR> [K)ill a Message]
To delete a message.

Y <CR> [Y)our Per. Mail]
To scan the message base for messages addressed to you.

TS <CR> [TX Txt Srch Msgs]
To find a text string in the message headers and message contents accessible to you.



GENERAL COMMANDS

General Bulletin Board Commands.

TYPE:

B <CR> [B)ulletin Listings]
To display the available Bulletins.

G <CR> [G)oodbye (Hang up)]
To terminate the Bulletin Board session.

H <CR> [H)elp Functions]
To receive Help Text on any command.

M <CR> [M)ode (Graphics)]
Toggles between the Graphics On/ Graphics Off Modes.

P <CR> [P)age Length Set]
To specify the number of lines the Bulletin Board displays before it prompts "More?"

X <CR> [X)pert On/Off]
Toggles between Expert On (no menus)/ Expert Off Modes.

V <CR> [V)iew Settings]
To display your Bulletin Board Mode settings.

W <CR> [W)rite User Info]
To change your password and other User Registration Information.

NEWS <CR> [NEWS file display]
To redisplay the text displayed when you initially logged on.

To provide customers with up-to-date information and an immediate response to questions, Xilinx provides a 24-hour electronic bulletin board. The Xilinx Technical Bulletin Board (XTBB) is available to all registered XACT customers. Users with full privileges can read files on the bulletin board, download those of interest to their own systems or upload files to the XTBB. They can also leave messages for other XTBB users.

New bulletin board users must answer a questionnaire when they first access the XTBB. After answering the questionnaire callers can browse through the bulletin and general information file areas. A caller with a valid XACT protection key or valid host ID will be given full user privileges within 24 hours.

The software and hardware requirements for accessing the Xilinx Technical Bulletin Board are:

Baud Rate	9600, 4800, 2400, or 1200 bps
Character Format	8 data bits, no parity, 1 stop bit
Phone Number	(408) 559-9327
Transfer Protocols	ASCII, Xmodem, (Checksum, CRC, 1K), Ymodem

Information contained on the XTBB is divided into three general categories: 1. Bulletins, 2. Files and 3. Messages.

1. Bulletins contain tidbits of up-to-date information; they can be displayed on-screen and can be downloaded.

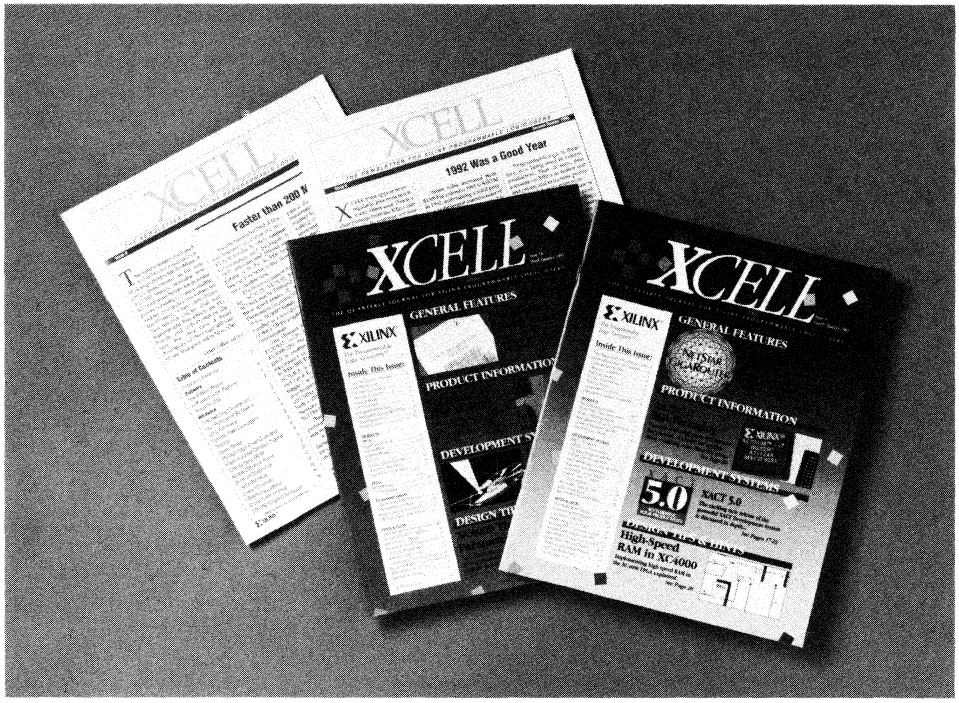
2. Files can contain just about anything (text, user programs, etc.). XTBB users with full privileges can download files to their own systems or upload files to the bulletin board.

3. Messages are used to communicate with other XTBB users; they can be general—available to everyone—or private.

The XTBB is based on a bulletin board system called PCBoard. This is a menu-driven system—you choose commands from menus to decide what happens next. To choose a menu command, simply type the highlighted first letter(s) of the command and press return <CR>. Listed below are some helpful hints for using the XTBB.

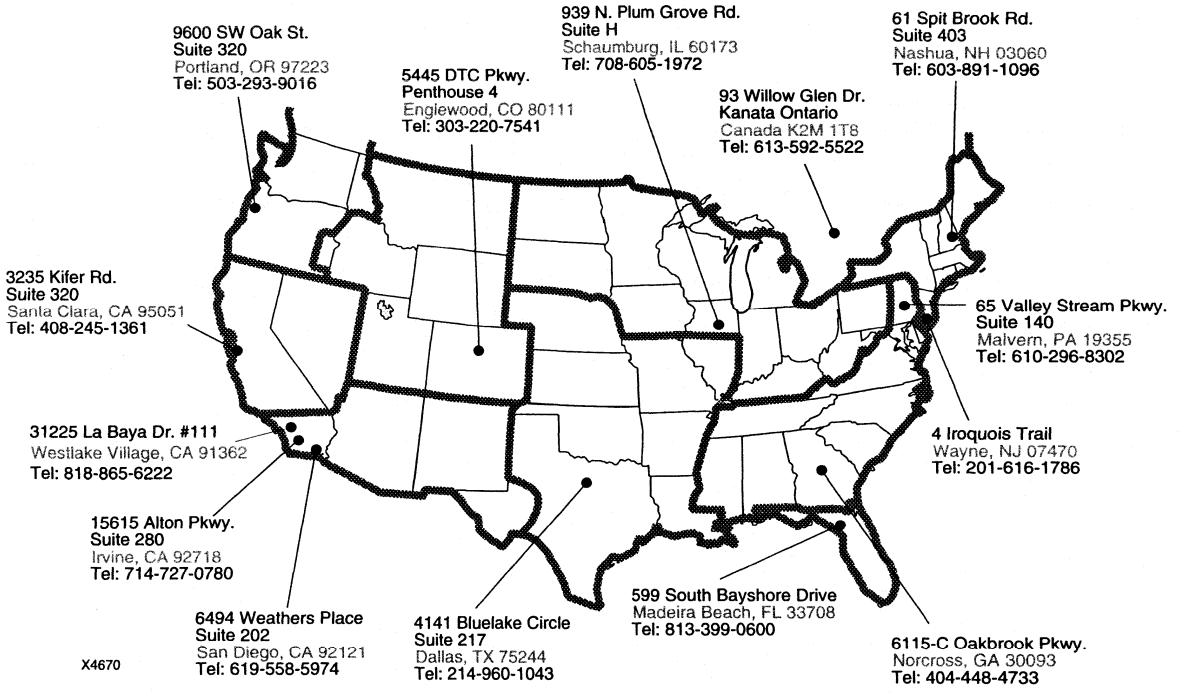
- To perform a sequence of commands, type the first letter of each command, followed by a space, and press return. For example, typing **F A <CR>** [**F**]ile Directories **A**]ll] sends you a listing of all file directories.

- The XTBB has an extensive help section. To get help, type **H <CR>** followed by the command in question. A short explanation of the command will be displayed. You can also type **H <CR>** inside a command, and get an explanation of the sub-commands.



XCELL, the quarterly customer newsletter, is dedicated to supplying up-to-date information to registered Xilinx customers, and is a valuable resource for systems designers. A typical issue of XCELL contains descriptions of recently-introduced products, updates on component and software availability and revision levels, applications ideas, design hints and techniques, and answers to frequently-asked technical questions.

Field Applications Engineers



North America

There are 18 Xilinx Field Applications Engineers in the locations shown above. Additional technical support is provided by Headquarters Applications.
Dial (800) 255-7778 for FPGAs.
Dial (408) 879-4686 for EPLDs.

The world-wide network of Xilinx Representatives and Distributors also gives technical support.

Europe

Each of the Xilinx European sales offices in England and Germany has resident Field Applications Engineers:
England (tel 44-932-349401);
Germany (tel 49-89-904 50 24 and 49-7940-58416).

Japan

Xilinx Japan is located in Tokyo and has resident Field Applications Engineers (tel 81-3-297-9191).

Asia

Xilinx Asia is located in Hong Kong and has a resident Field Applications Engineer (tel: 852-410-7240).

Programmable Logic Training Courses

Xilinx Programmable Logic Training Courses are comprehensive classes covering Xilinx components and development system products. All users of Xilinx products are encouraged to attend one of our Training Courses. Attending a Xilinx Training Course is one of the fastest and most efficient ways to learn how to design with programmable logic devices from Xilinx. Hands-on expert instruction with the latest information and software will allow you to implement your own designs in less time with more effective use of the devices.

Benefits

- Start or complete your design during the training class
- Reduce your learning time
- Make fewer design iterations
- Get to market faster
- Lower production costs
- Increase quality

Course Outline

The standard Xilinx Training Class lasts three days, with primary focus on the XC3000 and XC4000 families. All North American training sites teach the same class. The class covers the following topics:

Choosing a Device
Design Entry Methods
  Good Design Practices
Designing for the Xilinx Architecture
  Using Special Architecture Features
Xilinx Translation Flow
  Automatic Translation
  Debugging and Verifying Translation
  Incremental and Iterative Design Flow
Optimization of Designs
  Floorplanning
Choosing Configuration Options
  Configuration Debugging
Verifying Timing
  Simulation Flow

All classes include detailed lab exercises, with at least one computer for every two students. Classes are always kept up-to-date with the latest production products.

The classes are not specific to any particular design entry or verification tool. Viewlogic software is used as an example tool only, and is not taught in the class.

Training Options

Location	Tuition	Benefits
North America		
Xilinx Headquarters	$1000	Can meet with Xilinx applications engineers. Classes held twice per month
Rochester Institute of Technology, New York and University of Texas at Arlington, Texas	$1000	Workstations; emphasis on VHDL entry
Massachusetts Microelectronics Center, Massachusetts and Florida Atlantic University, Florida	$1000	East coast sites run by Xilinx headquarters
Distributor Locations	$450	Lower cost; local
Customer Site	Varies	Convenience; can focus on specific issues
International		
International Locations	Varies	Offered in over 12 countries; native language
Customer Site	Varies	Convenience; can focus on specific issues

Prerequisites

Students need only have a background in digital logic design. Basic familiarity with PCs and the DOS operating system is helpful but not required. Regional sites in New York and Texas offer workstation-based classes. The Advanced Training sessions require previous experience with Xilinx products.

Locations

Xilinx Headquarters

Classes are held twice per month at Xilinx headquarters in San Jose, California. During the class, you may elect to meet one-on-one with Xilinx Applications engineers to discuss specific issues not covered in the class. These may include using a specific third-party tool, optimizing your particular design, or more advanced issues beyond the coverage of the class. Also, Advanced Training sessions are typically held immediately following each class (see details below).

RIT and UTA

Sun workstations are used at training facilities at the Rochester Institute of Technology in Rochester, New York, and at the University of Texas at Arlington. If you are using a workstation, you will see the familiar interface if you attend a class at these sites. Also, classes at these sites focus more on high-level language entry (VHDL, Verilog), synthesis issues, and high-density design issues, since these tend to be more important to workstation users.

MMC and FAU

Xilinx runs classes at two other East Coast facilities, the Massachusetts Microelectronics Center in Westboro, Massachusetts, and Florida Atlantic University in Boca Raton, Florida. These facilities provide regular training classes for our East Coast-based customers.

Distributor Locations

Xilinx distributors sponsor training classes jointly with Xilinx, using the same material as in the headquarters classes. Since the distributor sponsors the class, the tuition cost is lower at $450. Locations include over three dozen cities across North America. Contact your local distributor, or Xilinx headquarters, for information on classes in your area.

International Locations

Xilinx classes are held throughout Europe, the Far East, and other international locations. Classes vary in length and tuition, but are based on the same material used in North America. Contact your local Xilinx sales office or representative for classes in your area.

Customer Site

Xilinx can bring its training class to your facility. A minimum number of ten students is preferred, but may be reduced depending on availability of equipment and instructors. Customer-site classes will naturally focus more on specific aspects of how your company is using Xilinx products. Customer-site classes may even be modified, including lengthening or shortening the class, to cover specific topics not normally covered in the standard class.

Advanced Training

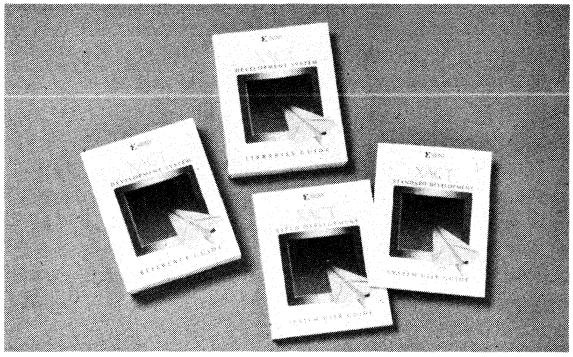
If you have already attended a Xilinx class, or have experience using Xilinx products, consider attending an Advanced Training session. Advanced Training classes, which last up to a full day, are offered at no charge to current Xilinx customers. Advanced Training sessions will vary according to the interests of the students. Popular topics include:

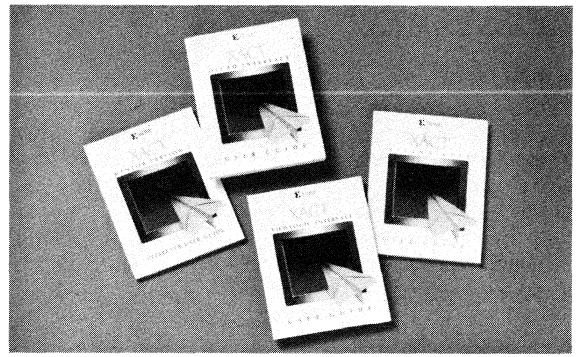
- Example Logic Design Techniques
- Timing Analysis and Avoiding Timing Hazards
- Using Optional Tools with Development System
- Design Methodology for Tough Designs
- Details of Advanced Optimization Capabilities

Advanced Training is held twice per month at Xilinx Headquarters. Advanced Training sessions may also be presented locally - contact your closest sales office for information.

Enrollment and Information

To enroll or to get information in North America, call the Xilinx Training Administrator at (408) 879-5090. Internationally please call your local Xilinx sales office.





Technical Literature

Xilinx provides manuals and supporting documents for development systems, libraries, CAE tool interfaces, and related software tools such as logic synthesis.These manuals are organized in several categories — Development System User Guides, Interface User Guides, Library Guides, Reference Guides and Hardware Guides.

Development System User Guides are introductory manuals that cover basic information about using Xilinx software.They address such topics as design entry and design verification in the Xilinx environment. User guides are provided for development system core software and for enhancements such as the Xilinx-ABEL and X-BLOX tools.

Interface User Guides address CAE tools as they relate to the Xilinx design environment. They address such topics as design entry and verification in the Xilinx environment using specific CAE tools. These guides include design flow, creating designs, translating designs into Xilinx format, verification and simulation of designs, and implementing designs. Tutorial information about the CAE tool is included in the interface guides, covering both design entry and design verification. When appropriate, sections covering CAE tool commands, options, and variables are also included in Interface User Guides.

Library Guides include information about primitives, gates, flip-flops, pads, I/O functions, and macros available for Xilinx programmable logic families. These guides includes appropriate symbols, descriptions, truth tables and schematics for design resource elements available across all Xilinx programmable logic device families. Functional selection guides list all elements available in each logic family.

Reference Guides cover details about each Xilinx software program, including the commands, options, variables, and arguments related to each program. These guides include information about the files required and the files generated, as well as warning and error messages. Reference guides address software functions and software capability, but do not always include "how to" information. Reference Guide contents are organized by function, following a "typical" design-flow model to provide details about specific functions that may be needed.

Hardware and Peripherals Guide covers hardware items such as the PROM Programmer, the XChecker Cable and Demonstrations Boards, and related software programs.

Documentation Sets

New documentation is provided in individual books covering development system software, CAE interfaces, libraries, and program reference information. Appropriate books are included with each software package. Additional books and book sets can be ordered.

CAE Tool Documents

Xilinx provides manuals covering CAE tools to those customers who buy these tools through Xilinx. These manuals are reprinted by Xilinx with permission from the CAE tool manufacturers. The content of these manuals is provided by the CAE tool manufacturers. Questions about the information in these manuals should be directed to the CAE tool manufacturer. The Viewlogic Workview Series I, Volumes 1, 2, and 3 books, and the Viewlogic ViewSynthesis book are examples of such manuals.

Overview

This section describes the Xilinx Automated CAE Tools (XACT) design environment for Xilinx FPGA and EPLD devices.

High-density programmable logic has created unique requirements for CAE software; the tools must deliver the ease-of-design and fast time-to-market benefits that have popularized FPGA and EPLD technologies, must be capable of implementing high-density logic designs on an engineer's desktop system, and must be easy-to-use and compatible with the user's existing design environment.

In order to meet those needs, Xilinx offers a variety of development system products optimized to support the Xilinx FPGA and EPLD architectures. Available products include state-of-the-art design compilation software, libraries and interfaces to popular schematic editors and timing simulators, and behavioral-based design entry tools. All Xilinx development system software is integrated under the Xilinx Design Manager (XDM), providing designers with a common user interface regardless of their choice of device architecture and tools. Supported platforms include the ubiquitous PC and several popular workstations.

As with other logic technologies, the basic methodology for FPGA design consists of three interrelated steps: entry, implementation, and verification. The design process is iterative, returning to the design entry phase for correction and optimization. Popular 'generic' tools are used for entry and simulation (for example, Viewlogic System's Viewdraw schematic editor and Viewsim simulator), but architecture-specific tools are needed for implementation.

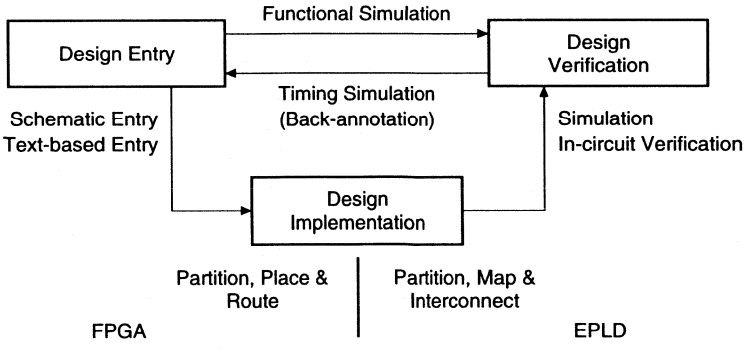
Design Entry

Schematic editors remain the most-popular tool for design entry. FPGA/EPLD symbol libraries and netlist interfaces are available for schematic editors such as Viewlogic's Viewdraw, OrCAD's SDT, and Mentor Graphics' Design Architect. These libraries reflect the wide variety of logic functions that can be implemented in FPGA/EPLD devices. For example, the library for the XC4000 family contains over 300 macros and primitives.

Schematic entry is well-suited to 'glue logic' designs, but is less effective for describing behavioral designs. Behavioral-oriented design entry methods, including Boolean equations and state-machine descriptions, are supported by the Xilinx-ABEL and X-BLOX products, as well as a number of products from CAE vendors such as Data I/O, MINC, Logical Devices, and ISDATA.

As the density and complexity of FPGA designs increase to 10,000 gates and beyond, gate-level entry tools often are cumbersome, and the use of logic synthesis and high-level description languages (HDLs), such as VHDL, can raise designer productivity. The use of HDLs requires synthesis tools that effectively compile designs for the target FPGA architecture. Xilinx offers interfaces for synthesis tools from Synopsys and Viewlogic Systems. Other CAE vendors, such as Mentor Graphics, Cadence Design Systems, and Exemplar Logic, also offer synthesis tools tailored for the Xilinx device architectures.

Many engineers prefer visually oriented design-entry techniques over text-based HDLs. The benefits of HDLs are provided to these designers with tools that provide high-level design constructs in a symbolic format compatible with graphics-based schematic editors. X-BLOX is a graphics-based high-level language that allows designers to use a schematic editor to enter designs as a set of generic modules. The X-BLOX compiler optimizes the modules for the target device architecture, automatically choosing the appropriate architectural resources for each function.



The XACT design environment supports hierarchical design entry, with top-level drawings defining the major functional blocks, and lower-level descriptions defining the logic in each block. The implementation tools automatically combine the hierarchical elements of a design. Different hierarchical elements can be specified with different design entry tools, allowing the use of the most convenient entry method for each portion of the design. In this type of 'mixed-mode' design entry, designers can intermix schematic, text, gate-level and behavioral-level design, permitting the re-use of previously designed modules and easing the transition to higher-level design methodologies.

Design Implementation

After the design is entered, implementation tools map the logic into the resources of the target device architecture, determine an optimal placement of the logic, and select the routing channels that connect the logic and I/O blocks. Xilinx design implementation tools apply a very high degree of automation to these tasks. A design compilation utility, XMAKE, automatically retrieves the design's input files and performs all the necessary steps to create the EPLD or FPGA configuration program.

Although the automated tools are very efficient, no tool can understand a design as well as the designer. For demanding applications, the user can exercise various degrees of control over the automated implementation process. Optionally, user-designated partitioning, placement, and routing information can be specified as part of the design entry process (typically, right on the schematic). The implementation of highly structured designs can greatly benefit from the basic floorplanning techniques familiar to designers of large gate arrays.

For Xilinx FPGAs, the automatic tools are complemented by an interactive, graphics-based editor that allows users to view and manipulate a model of the logic and routing resources inside the FPGA device, providing the user with visibility into the implementation of the design.

Design Verification

Verification of FPGA/EPLD designs typically involves a combination of in-circuit testing, simulation, and static timing analysis. The user-programmable nature of these devices allows designs to be tested immediately in the target application. For Xilinx FPGAs, download cables are provided that allow for the direct downloading of a bitstream from a PC or workstation to an FPGA device on a target board. Demonstration/prototyping boards are also available. The implementation tools include timing calculators to determine the post-layout timing of implemented designs, thereby supporting timing simulation. Back-annotation – the process of mapping the timing information back into the signal names and symbols of the schematic – eases the debugging effort. A static timing analyzer can be used to examine a design's logic and timing to calculate the performance along signal paths, identify possible race conditions, and detect set-up and hold-time violations. Timing analyzers do not require the user to generate input stimulus patterns or test vectors.

Xilinx software is available both in bundled packages containing entire sets of tools and as separate products. New enhancements are constantly being developed, and update services are available to ensure timely access to the latest versions.

Xilinx Software Available on CD-ROM

Starting in early 1994, Xilinx software and updates are available on CD-ROM for the PC and for most workstations (Sun and HP 9000 / 700 Series). Here are some of the benefits :

- Faster Installation: No more wasted time, feeding floppy after floppy into the PC. No more waiting for workstation tapes to spin, looking for the proper data. Installing or updating Xilinx software is as easy as popping in one CD-ROM disk.
- Software Compatibility: With CD-ROM, all Xilinx software for all platforms is on the same disk. New install utilities monitor the software configuration, ensure executable version compatibility, and update only the necessary files to keep the software up-to-date. Archiving old versions of XACT software is as easy as storing one CD-ROM disk.
- Future Improvements: Xilinx will eventually add on-line documentation, tutorials, application notes, and product demonstrations.

To start getting your updates on ISO 9660 standard CD-ROM, simply FAX a letter to Xilinx customer service at (408) 559-0115, indicating the serial numbers of each of the products you'd like to receive on CD-ROM.

XACT 5.0 Accelerating Your Productivity

The newest version of the XACT Development System, XACT 5.0, starts shipping in the first quarter of 1994. Registered Xilinx development system owners with active software maintenance agreements receive this update automatically.

XACT 5.0 contains several enhancements and features that increase design productivity.

Enhancements Common to All Devices

- Windows Compatibility

For PC users, XACT 5.0 offers compatiblity with Windows v3.1. A new DOS extender allows designers to run the XACT software without exiting Windows. PPR/APR can be run in the background/multi-tasking mode while the computer is used for other applications.

- Common Flows for All Xilinx Architectures

XACT 5.0 takes a major step towards the standardization of the Xilinx design environment. Schematic and behavioral entry is available for both FPGAs and EPLDs. Popular third-party tools (including schematic editors, logic compilers, fitters, and simulators) support all Xilinx families. FPGA and EPLD development uses the same basic design flow and supports the Unified Library.

- Unified Libraries

The Xilinx libraries have been re-engineered for easy migration between Xilinx programmable logic families. With the exception of a few architectural dependent symbols, designs utilizing the Unified Library can easily be retargeted to any Xilinx FPGA or EPLD device.

- CD-ROM Software Distribution and Programmable Key

Xilinx now offers most software on CD-ROM. CD-ROM distribution of software combined with a programmable key (the key is for PCs only) offer the following capabilities:

  - The Programmable Key lets the user test drive any Xilinx program before deciding to buy it.
  - Installation is easier with all software on a single CD-ROM instead of multiple floppy disks.

- New Design-Rule Checker

XACT 5.0 contains a new design rule checker to help pinpoint problem areas before the design gets to the place-and-route phase. The output of the design rule checker contains detailed descriptions of problems, including specific signal and symbol names.

- Xilinx-ABEL improvements

Xilinx-ABEL 5.0 includes several enhancements that make it easier to design sections of an FPGA or EPLD, or to design an entire EPLD.

- Simulation Make Command

Just as XMAKE automatically performs all the steps necessary to implement a design, a new utility called XSimMake automatically performs all the steps needed to prepare a completed FPGA or EPLD design for full timing simulation, including the back-annotation of signal names.

XC4000 Software Enhancements

- Incremental Design

The XACT 5.0 release adds incremental design capability for XC4000-series designs. The incremental design methodology utilizes a previous design as a guide for the new design. For small changes, incremental design can significantly shorten the overall design cycle.

- PPR Improvements

XACT 5.0 improves PPR routability and device performance. The improvements to the place-and-route algorithms are particularly noticeable in the implementation of highly-structured designs.

- Enhanced XACT-Performance Control

This release continues to build on the features of XACT-Performance, like management of clock skew, support of the RPM library elements, RAM support, and improved carry-logic support.

- Improved Back-Annotation for Functional and Timing Simulation

Users who have experienced difficulty in simulating XC4000 designs will find the XACT 5.0 release significantly easier.

XC3000 Software Enhancements

- PPR for the XC3000A/XC3100A

Design implementation of XC3000A family FPGAs is now supported by the same PPR (Partition, Place, and Route) program used for the XC4000 family. This brings the features of PPR, including XACT-Performance and X-BLOX support, to the XC3000A/XC3100A architecture. (The XC3000 and XC3100 families will continue to be supported by the APR implementation program.)

- XACT-Performance for XC3000A/XC3100A

XACT-Performance allows designers to enter their performance requirements, up-front, at the schematic level. Previously available for the XC4000 family only, XACT 5.0 extends this capability to the XC3000A/XC3100A families.

- X-BLOX for XC3000A/XC3100A

In the XACT 5.0 release, X-BLOX now supports the XC3000A and XC3100A families.

Bundled Packages

Xilinx Development System software is available as bundled packages or separate products. Packages are designed to address the needs of different types of users and are available for a variety of CAE systems and platforms.

Packages consist of Core implementation tools and Libraries and Interface. Packages offer discounts over individual software products purchased separately.

Base packages provide schematic capture and simulation interfaces, design implementation tools and download hardware for low-complexity Xilinx devices. These devices include the XC7200/XC7300 EPLD families (PC, Sun and HP only), the complete XC2000 FPGA family, XC3000 (up to XC3x42) and XC4000 (up to XC4003) FPGA families of devices. A special *Stand–alone* version is available for Viewlogic on the PC that includes the ViewDraw schematic editor and ViewSim simulator (limited to 5,000 gates).

Standard packages provide schematic capture and simulation interfaces, design implementation tools and download hardware for all of Xilinx devices. These devices include the XC7200/XC7300 EPLD family (PC, Sun and HP only) and the complete XC2000, XC3000/XC3100 and XC4000 FPGA families. In addition the X-BLOX module generator and optimizer is provided. A special *Stand-alone* version is available for Viewlogic on the PC that includes the ViewDraw schematic editor and ViewSim simulator (unlimited gates).

Extended packages provide all the capability of the Standard package plus more, and is offered for the Viewlogic /S. This special *Stand-alone* version is available for Viewlogic on the PC that includes the Standard Viewlogic Package, ViewDraw schematic editor, ViewSim simulator (unlimited gates) and ViewSynthesis.

The Viewlogic *Stand-alone* systems provide a complete turn-key environment for designing EPLDs and FPGAs.

Individual Products

All of the Xilinx Development System software, hardware and documentation is available as individual products for adding to an existing package or creating a new one.

Libraries and Interface – Contains symbol libraries for specified schematic editor, simulation models with timing information for specified simulator and programs to translate between the schematic editor or simulator's file format and the XNF file format.

Core Implementation – Provides the software necessary to process an XNF file into a BIT or PROM file that can be downloaded into a Xilinx device. Includes tools for logic reduction, design rule checking, mapping, automatic placement and routing, bitstream generation and PROM file generation.

X-BLOX Module Generator + Optimizer – Allows block diagrams to be entered using a familiar schematic editor. Using built-in expert knowledge, X-BLOX software automatically optimizes the design to take full advantage of the unique features of the XC3000A and XC4000 FPGA families.

Xilinx ABEL – Supports text based design entry and netlist translation using ABEL high level description language. ABEL language supports different design styles including Boolean equations, truth tables and encoded or symbolic state machines.

Parallel Download Cable – Supports downloading of bitstream and PROM files from the parallel port of IBM PCs and compatibles.

XChecker Cable – Supports downloading of bitstream and PROM files and readback of configuration data and internal node values. This cable uses the serial port of IBM PCs & compatibles and Sun & HP workstations.

XCFPGA Demoboard – Provides demonstration or prototype capability for XC2000, XC3000/A, XC3100/A devices in 68-pin PLCC packages and XC4000 family devices in 84-pin PLCC. This board is combined to offer flexibility for learning and prototyping.

6-Packs – These are specially priced multiple software packages to allow for quantity purchase discounts. Note: they are not available for all development system products.

Xilinx Automatic CAE Tools Product Overview

FPGA Design Flow

The Xilinx Automatic CAE Tools (XACT Development System) use a 3-step design process:

- Step 1 Design Entry
- Step 2 Design Implementation
- Step 3 Design Verification

The Xilinx Logic Libraries and XNF Interface Products support design entry with popular schematic logic drawing systems supplied by multiple vendors, providing easy entry to the XACT Development System.

Logic synthesis, partitioning, and optimization programs translate the design specifications into CLBs and IOBs unique to the LCA architecture. Subsequent programs perform automatic placement and routing to complete the LCA design.

While completely automatic implementation is desirable for both low- and high-complexity designs, the designer may prefer an interactive process, especially in high-performance designs. This interactive editing can range from rerouting a few previously automatically routed nets, to prerouting critical nets or preplacing CLBs prior to design completion using APR/PPR, to more extensive control over logic partitioning and placement into CLBs. The Design Implementation software gives the designer an option for direct control over specific logic mapped into CLBs (partitioning) to provide better distribution of logic signal routing through the LCA device. The XACT Design Editor, XDE, is extremely versatile, ranging from design entry to CLB and signal routing manipulations. This combination of automatic and interactive design editing capability is a unique feature provided by Xilinx.

Logic simulation or actual in-circuit emulation provides for functional verification, while timing analysis permits verification of critical timing paths under worst-case conditions. The system contains a compiler to generate bitstream patterns to configure the LCA device according to the designer's specification. The overall design flow is illustrated on page 7-6.

An important feature of the XACT Development System is the capability to incorporate design changes, frequently encountered during verification. Small changes can be made to the schematics and then automatically incorporated into the existing design with minimal impact on existing routing and performance. Using this "incremental design" capability, the designer can develop "production quality" programmable gate arrays on a PC or engineering workstation.

EPLD Design Flow

The Xilinx XEPLD development tool also uses a 3-step design process.

Step 1 – Design Entry

Step 2 – Design Implementation

Step 3 – Design Verification

Designs can be represented as schematics consisting of XEPLD library components, as behavioral descriptions, or a mixture of both.

The XEPLD translator reads Boolean equations and schematic netlists. It minimizes equations, optimizes the design and maps the result onto a selected EPLD device.

The DS-550 XEPLD translator produces a simulation-model file for either the Viewlogic Viewsim or OrCAD VST simulator. The overall design flow is illustrated on page 7-8.

Platform and Environment Support

The Xilinx Automatic CAE Tools, XACT, are currently available for the following platforms:

- '386/'486 PCs, PS/2, and compatibles
- HP700 Series
- Sun-4 and SparcStation Series

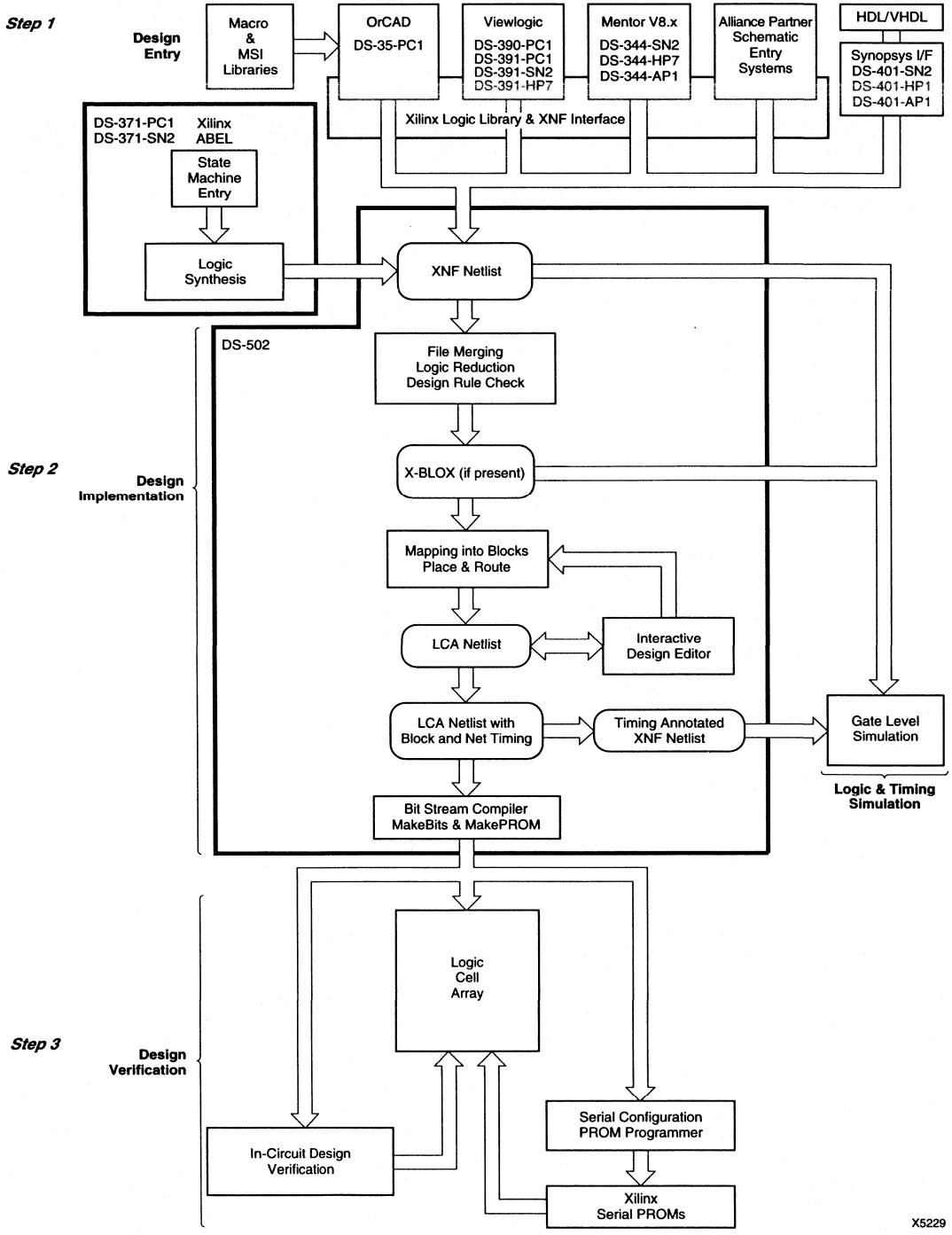
Xilinx and third-party vendors have developed library and interface products compatible with a variety of design entry and simulation environments. Xilinx has provided a standard interface file specification, XNF, to simplify file transfers into and out of the XACT Development System.

Xilinx directly supports the following design environments:

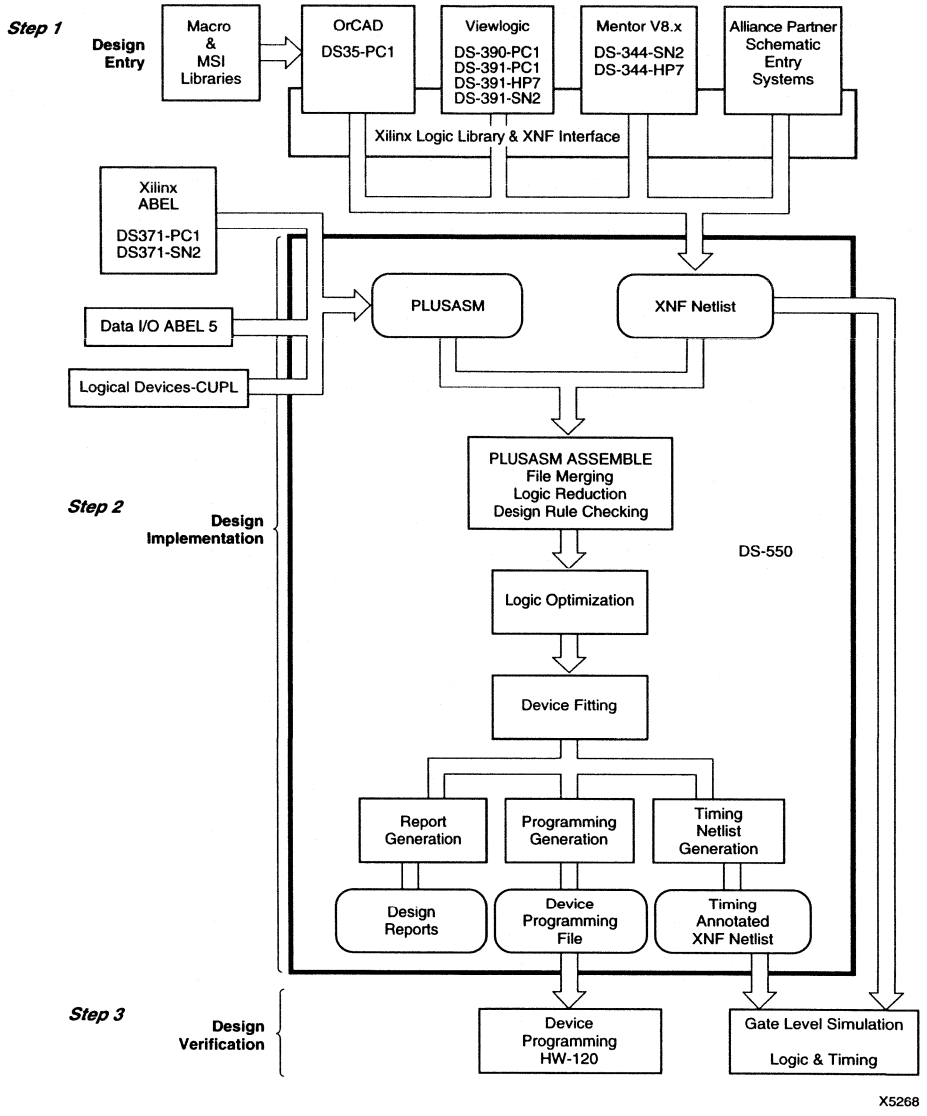
- Viewlogic Viewdraw and Viewsim
- Mentor Graphics Design Architect and Quicksim II
- OrCAD SDT and VST

Several other environments are supported by third-party vendors.

The XACT Design Manager, XDM, simplifies the selection of command-line options with pull-down menus and online help text. Application programs ranging from schematic capture to APR/PPR can be accessed from the XDM, while the sequence of program commands is generated and stored for documentation prior to execution. The XMAKE command in the XDM automates the entire translation, optimization, merging, and mapping process.



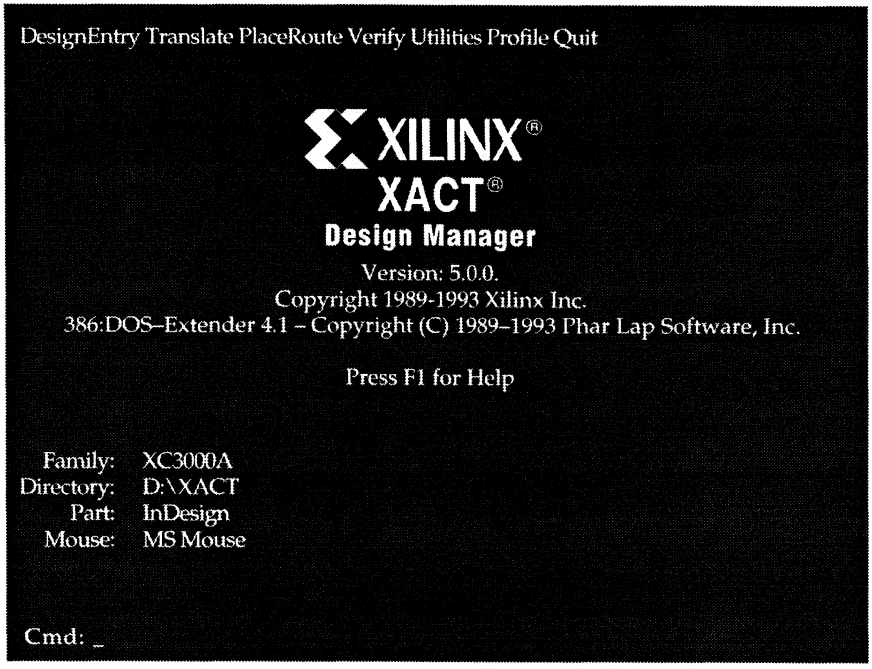
FPGA Design Flow



EPLD Design Flow

The Xilinx Design Manager—Simplifies the Design Flow

- Permits running all Xilinx software from menus
- Automates design translation via XMake facility
- Provides on-line help for all menus, programs and options



X5269

XMake Command

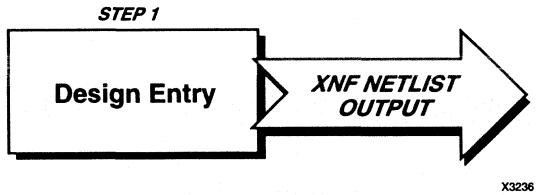
- Automatically invokes all other translation programs as required to compile a design into an FPGA or EPLD
- Supports hierarchically structured designs

Extensive On-line Help

The Design Manager contains on-line Help for

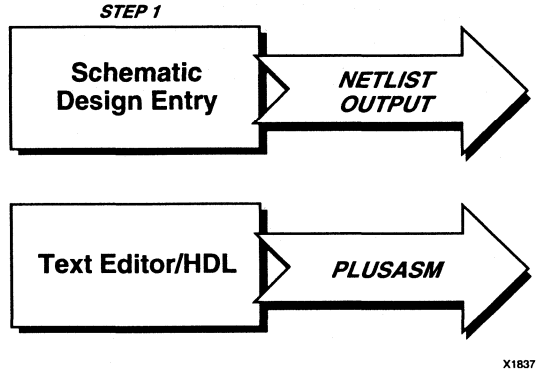
- Every menu
- Every program
- Every program option
- Design-flow suggestions

FPGA Design Flow



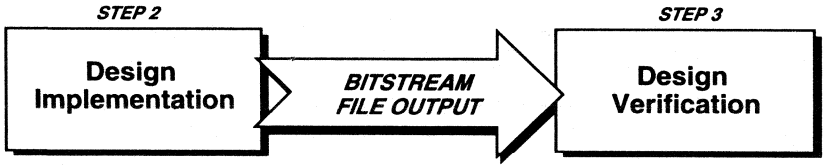
- Open development system supports design entry and simulation on popular CAE systems
- Interfaces available from Xilinx for PC- and workstation-based environments:
  - Viewdraw, OrCAD, Mentor V8
  - XACT-Performance allows designers to enter their design performance requirements directly in their schematics
- Standard macro library includes over 300 elements
- Several other PC and workstation environments are supported by third-party vendors
- Xilinx ABEL provides efficient state machine implementation for LCA architecture
- Synthesis from behavioral hardware description languages (HDLs) to FPGA device with interfaces to Synopsys and Viewlogic
- X-BLOX libraries for all supported schematic capture interfaces. X-BLOX module generator and optimizer produces high-performance designs quickly

XEPLD Design Flow



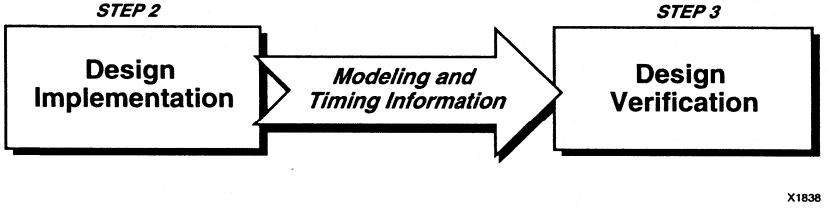
- Open development system supports design entry and simulation on popular CAE systems.
- Interfaces available from Xilinx for PC-and Workstation-based environments:
  - Viewlogic, OrCAD, Mentor Graphics V8 for schematic Entry
  - XABEL for state machine, boolean equation entry
  - PLUSASM for Xilinx proprietary boolean equation entry
- Cadence Design, Commercial PLD compilers: ABEL, CUPL, LOG/iC design environment are supported by third party vendors
- Built-in automatic PAL conversion process

FPGA Design Flow (continued)



- Complete system translates design into programmable gate arrays
- Performance based on specified timing requirements
- Partitions gate-level design logic into LCA architecture (CLB/IOB)
- Automatic logic reduction and partitioning removes unused logic, e.g., unused counter outputs
- X-BLOX module generator software optimizes design for LCA architecture
- Interfaces available from Xilinx to popular simulators for logic and full timing simulation
  - Mentor Graphics
  - Viewlogic
  - OrCAD
  - Synopsys
- Several other simulators are supported by third-party vendors
- LCA user-programmability permits real time, in-circuit debugging
- XChecker download cable allows the LCA device to be programmed in-circuit during debugging

XEPLD Design Flow (continued)



- Complete system translates design into Xilinx EPLD devices
- Interfaces available from Xilinx to popular simulators for logic/full timing simulation:
  - Mentor Graphics, Cadence-Verilog, OrCAD, Viewlogic
- Board-level simulation support through Logic Modeling Device model generation
- Predictable design performance independent of physical placement and routing or device utilization
- High-speed design compilation for quick design iteration
- Automatic mapping and logic optimization, but also complete design control
- Automatic FuseMap generation

Packages for the PC

Feature	Viewlogic Base	Viewlogic Standard	Viewlogic *Stand-alone* Base	Viewlogic *Stand-alone* Standard	Viewlogic *Stand-alone* Extended	OrCAD Base	OrCAD Standard
Libraries and Interface	✓	✓	✓	✓	✓	✓	✓
Schematic Editor			✓	✓	✓		
Simulator (Limited Gates)			✓				
Simulator (Unlimited Gates)				✓	✓		
EPLD Devices	✓	✓	✓	✓	✓	✓	✓
FPGA Devices (Limited) (1)	✓		✓			✓	
FPGA 2K, 3K, 4K		✓		✓	✓		✓
Core Implementation	✓ (2)	✓	✓ (2)	✓	✓	✓ (2)	✓
Synthesis Tools		✓		✓	✓		✓
X-BLOX					✓		
Parallel Download Cable			✓			✓	
XChecker Cable	✓	✓		✓	✓		✓
XCFPGA Demonstration Board	✓	✓	✓	✓	✓	✓	✓
Hotline Support	✓ (3)	✓	✓	✓	✓	✓ (3)	✓
1 Year Software Updates		✓	✓	✓	✓		✓

X5251

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003
(2) Base Development System does not include XDE
(3) $995 OrCAD and Viewlogic Base Packages only have six month phone support

Packages for Workstations

Feature	Viewlogic – Standard	Mentor – Standard	Synopsys – Standard	Cadence (2)
Libraries and Interface	✓	✓	✓	✓
Schematic Editor				
Simulator Limited Gates)				
Simulator (Unlmited Gates)				
EPLD Devices	✓	✓		
FPGA Devices (Limited) (1)				
FPGA 2K, 3K, 4K	✓	✓	✓	✓
Core Implementation	✓	✓	✓	✓
Synthesis Tools				
X-BLOX	✓	✓	✓	✓
Parrallel Download Cable				
XChecker Cable	✓	✓	✓	✓
XCFPGA Demonstration Board	✓	✓	✓	✓
Hotline Support	✓	✓	✓	✓
1 Yr Software Updates	✓	✓	✓	✓

X5252

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003
(2) Availability through Cadence

Individual Products for the PC and Workstation

Individual Products	PC-1	Workstation		
		SN2	AP1	HP7
Viewlogic Interface (DS-391)	✓	✓		✓
Viewlogic Schematic Editor (DS-390)	✓			
Viewlogic Simulator (DS-290)	✓			
OrCAD Interface (DS-35)	✓			
Mentor V7 FPGA Interface (DS-343)			✓	
Mentor V8 Interface (DS-344)		✓	✓	✓
Cadence FPGA Interface		✓		✓
Synopsys FPGA Interface (DS-401)		✓	✓	✓
FPGA Core Implementation (DS-502)	✓	✓	✓	✓
X-BLOX (DS-380)	✓	✓	✓	✓
Xilinx ABEL (DS-371)	✓	✓		
Parallel Download Cable	✓ (1)			
XChecker Cable	✓ (2)	✓	✓	✓
FPGA Demonstration Board	✓	✓	✓	✓
EPLD Core Implementation (DS-550)	✓	✓		✓

x5253

Packages are made up of some of the components above, please see Software Package section for supported platforms.

Note: SN2 = Sun-4, AP1 = Apollo/HP400, HP7 = HP 700 series
(1) Only included with $995 Base package
(2) XChecker Cable not included with $995 OrCAD or Viewlogic Base Package

Order Codes

Order codes for development-system products consist of a multiple-field part number. The first field indicates the product category. Additional fields indicate the third-party CAE vendor for interface tools, the package name or individual product number and the platform.

For example, the following order code indicates the category as *Development System*, the interface CAE vendor as *Viewlogic*, the package as *Standard*, the platform as *IBM PC* or compatible, and the media as *CD-ROM*.

DS-VL-STD-PC1-C

To order 6-PACKS of a product (where applicable), append a -6P to the end of the order code. For example, the following order code indicates a *6-PACK* of the *Standard* package for *Viewlogic* on the *PC* with CD-ROM media.

DS-VL-STD-PC1-C-6P

The following table shows valid product category, CAE vendor, package type, platform and media type codes.

Product Category	Code
Development System	DS
Support and Updates	SC
Hotline	HL
Re-instate Updates	SR
Product Upgrade	DX
Documentation	DM
Hardware	HW
Training Course	TC
Documentation Updates	DP
Interface Vendor	
OrCAD	OR
Viewlogic	VL
Viewlogic *Stand-alone*	VLS
Mentor, version 7	MN7
Mentor, version 8	MN8
Synopsys	SY
Package Type	
Base System	BAS
Standard System	STD
Extended System	EXT
Platform	
IBM PC or compatible	PC1
Sun-4	SN2
Apollo or HP 400	AP1
HP700	HP7
IBM RS6000	RS6
DEC Alpha	DA1
NEC	NC2
Media type	
CD-ROM	C
3.5" Floppy Disk (for PC only)	3

Bundled Packages Product Descriptions

Table of Contents

OrCAD – Base System (PC)

Base System Includes:

- Schematic Interface for OrCAD SDT386+ with library support for XC2000, XC3000/XC3100 and XC4000 FPGAs and XC7200 and XC7300 EPLDs
- Functional and Timing Simulation Interface for OrCAD VST386+
- Core implementation software for EPLDs and FPGAs with device support for XC7000, XC2000, XC3000 up to XC3042/XC3142 and XC4000 up to XC4003
- FPGA Demonstration Board
- Parallel Download Cable
- Six month Hotline Support

Notes:

- Hotline Telephone Support is included for the first six months only. Additional Hotline Support or Updates may be purchased separately.
- This package does not include the OrCAD SDT schematic capture or VST simulation tools. They must be purchased separately from OrCAD.
- XDE-Xilinx Design Editor is not included

Hotline Support Includes:

- Hotline Telephone Support (six months)
- Access to Xilinx bulletin board
- Apps FAX

Required Hardware Environment:

- Fully compatible PC386/486
- MS-DOS version 5.0 (minimum)
- Minimum 60 Mbyte hard-disk space for Xilinx software
- One 3.5" High-Density floppy disk drive
- VGA display
- One parallel and two serial ports
- 8 Mbyte of RAM
- Mouse

Package Features - OrCAD PC

Feature	Base	Std.
Libraries and Interface	✓	✓
Schematic Editor		
Simulator (Limited)		
Simulator (Unlimited)		
EPLD Devices	✓	✓
FPGA Limited (1)	✓	
FPGA 2K, 3K, 4K		✓
Core Implementation	✓(2)	✓
Synthesis Tools		
X-BLOX		✓
Parallel Download	✓	
XChecker Cable		✓
FPGA Demo Board	✓	✓
Hotline Support	✓	✓
1 Yr Software Update		✓

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003

(2) XDE Design Editor not included

X5235

OrCAD – Standard System (PC)

Standard System Includes:

- Schematic Interface for OrCAD SDT386+ with library support for XC2000, XC3000/XC3100, and XC4000 FPGAs and XC7200 and XC7300 EPLDs
- Functional and Timing Simulation Interface for OrCAD VST386+
- X-BLOX Module Generator + Optimizer
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC2000, XC3000/XC3100, and XC4000
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for first year

Note:

- This package does not include the OrCAD SDT schematic capture or VST simulation tools. They must be purchased separately from OrCAD.

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

- Fully compatible PC386/486
- MS-DOS version 5.0 (minimum)
- Minimum 60 Mbyte hard-disk space for Xilinx software
- One 3.5" High-Density floppy disk drive or ISO 9660 compatible CD-ROM drive
- VGA display
- One parallel and two serial ports
- 8 Mbytes of RAM for devices up to XC4006
- 16 Mbytes of RAM for XC4008 and above
- Mouse

Package Features - OrCAD PC

Feature	Base	Std.
Libraries and Interface	✓	✓
Schematic Editor		
Simulator (Limited)		
Simulator (Unlimited)		
EPLD Devices	✓	✓
FPGA Limited (1)	✓	
FPGA 2K, 3K, 4K		✓
Core Implementation	✓(2)	✓
Synthesis Tools		
X-BLOX		✓
Parallel Download	✓	
XChecker Cable		✓
FPGA Demo Board	✓	✓
Hotline Support	✓	✓
1 Yr Software Update		✓

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003

(2) XDE Design Editor not included

X5236

Viewlogic – Base System (PC)

Base System Includes:

- Schematic Interface for ViewDraw with library support for XC2000, XC3000/XC3100 and XC4000 FPGAs and XC7200 and XC7300 EPLDs
- Functional and Timing Simulation Interface for ViewSim
- Core implementation software for EPLDs and FPGA with device support for XC7000, XC2000, XC3000 up to XC3042/XC3142 and XC4000 up to XC4003
- FPGA Demonstration Board
- Parallel Download Cable
- Six months Hotline Support

Notes:

- This package includes Hotline Support for the first six months only. Additional Hotline Support and Updates can be purchased.
- This package does not include ViewDraw schematic capture or ViewSim simulation tool. They must be purchased separately from Viewlogic or Xilinx (see *Stand-alone* packages).
- XDE-Xilinx Design Editor – not included
- Interface and libraries support Workview 4.1, Workview PLUS, and PRO series

Hotline Includes:

- Hotline Telephone Support (six months)
- Access to Xilinx bulletin board
- Apps FAX

Required Hardware Environment:

- Fully compatible PC386/486
- MS-DOS version 5.0 (minimum)
- Minimum 60 Mbytes disk space
- One 3.5" High-Density floppy drive
- VGA display
- Mouse
- One parallel and two serial ports
- 8 Mbytes of RAM

Package Features - Viewlogic PC

Feature	VL Base	VL Std.	VLS Base	VLS Std.	VLS Ext.
Libraries and Interface	✓	✓	✓	✓	✓
Schematic Editor			✓	✓	✓
Simulator (Limited)			✓		
Simulator (Unlimited)				✓	✓
EPLD Devices	✓	✓	✓	✓	✓
FPGA Limited (1)	✓		✓		
FPGA 2K, 3K, 4K		✓		✓	✓
Core Implementation	✓(2)	✓	✓(2)	✓	✓
Synthesis Tools					✓
X-BLOX		✓		✓	✓
Parallel Download	✓				
XChecker Cable		✓	✓	✓	✓
FPGA Demo Board	✓	✓	✓	✓	✓
Hotline Support	✓	✓	✓	✓	✓
1 Yr Software Update		✓	✓	✓	✓

Note: VL = Viewlogic, VLS = Viewlogic *Stand-Alone*

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003

(2) XDE Design Editor not included

X5237

Viewlogic – Standard System (PC)

Standard System Includes:

- Schematic Interface for ViewDraw with library support for XC2000, XC3000/XC3100 and XC4000 FPGAs and XC7200 and XC7300 EPLDs
- Functional and Timing Simulation Interface for ViewSim
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC7200, XC3000/XC3100 and XC4000
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for the first year

Note:

- This package does not include ViewDraw schematic capture or ViewSim simulation tools. They must be purchased separately from Viewlogic or Xilinx (see *Stand-alone* packages).
- Interface and libraries supports Workview 4.1, Workview PLUS, and ProSeries

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

- Fully compatible PC386/486
- MS-DOS version 5.0 (minimum)
- Minimum 80 to 100 Mbytes hard-disk space
- One 3.5" or 5.25" High-Density floppy disk drive or ISO 9660 compatible CD-ROM drive
- VGA display
- Mouse
- One parallel and two serial ports
- 8 Mbytes of RAM for devices up to XC4006
- 16 Mbytes of RAM for XC4008 and above

Package Features - Viewlogic PC

Feature	VL Base	VL Std.	VLS Base	VLS Std.	VLS Ext.
Libraries and Interface	✓	✓	✓	✓	✓
Schematic Editor			✓	✓	✓
Simulator (Limited)			✓		
Simulator (Unlimited)				✓	✓
EPLD Devices	✓	✓	✓	✓	✓
FPGA Limited (1)	✓		✓		
FPGA 2K, 3K, 4K		✓		✓	✓
Core Implementation	✓(2)	✓	✓(2)	✓	✓
Synthesis Tools					✓
X-BLOX		✓		✓	✓
Parallel Download	✓				
XChecker Cable		✓	✓	✓	✓
FPGA Demo Board	✓	✓	✓	✓	✓
Hotline Support	✓	✓	✓	✓	✓
1 Yr Software Update		✓	✓	✓	✓

Note: VL = Viewlogic, VLS = Viewlogic *Stand-Alone*

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003

(2) XDE Design Editor not included

X5238

Viewlogic *Stand-alone* – Base System (PC)

Stand-alone Base System Includes:

- ViewDraw Schematic Editor with library support for XC2000, XC3000/XC3100, XC4000 FPGAs and XC7200 and XC7300 EPLDs
- ViewSim Functional and Timing Simulation for designs up to 5,000 gates
- Core implementation software for EPLDs and FPGAs with device support for XC7000, XC2000, XC3000 up to XC3042/XC3142 and XC4000 up to XC4003
- FPGA Demonstration Board
- Parallel Download Cable

Note:

- The Viewlogic software (ViewDraw, ViewSim) included in this package is the DOS-based Workview 4.1 software

Support and Updates include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

- Fully compatible PC386/486
- MS-DOS version 5.0 (minimum)
- Minimum 70 Mbytes hard-disk space
- One 3.5" High-Density floppy disk drive or ISO 9660 compatible CD-ROM drive
- VGA display
- Mouse
- One parallel and two serial ports
- 8 Mbytes of RAM

Package Features - Viewlogic PC

Feature	VL Base	VL Std.	VLS Base	VLS Std.	VLS Ext.
Libraries and Interface	✓	✓	✓	✓	✓
Schematic Editor			✓	✓	✓
Simulator (Limited)			✓		
Simulator (Unlimited)				✓	✓
EPLD Devices	✓	✓	✓	✓	✓
FPGA Limited (1)	✓		✓		
FPGA 2K, 3K, 4K		✓		✓	✓
Core Implementation	✓(2)	✓	✓(2)	✓	✓
Synthesis Tools					✓
X-BLOX		✓		✓	✓
Parallel Download	✓				
XChecker Cable		✓	✓	✓	✓
FPGA Demo Board	✓	✓	✓	✓	✓
Hotline Support	✓	✓	✓	✓	✓
1 Yr Software Update		✓	✓	✓	✓

Note: VL = Viewlogic, VLS = Viewlogic *Stand-Alone*

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003

(2) XDE Design Editor not included

X5239

Viewlogic *Stand-alone* – Standard System (PC)

Stand-alone Standard System Includes:

- ViewDraw Schematic editor with library support for XC2000, XC3000/XC3100, XC4000 FPGAs and XC7200 and XC7300 EPLDs
- ViewSim Functional and Timing Simulation for designs (unlmited gates)
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC2000, XC3000/XC3100, and XC4000
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for the first year

Note:

- The Viewlogic software (ViewDraw, ViewSim) included in this package is the DOS-based Workview 4.1 software

Support and Updates Include:

- Hotline Telephone support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

- Fully compatible PC386/486
- MS-DOS version 5.0 (minimum)
- Minimum 90 to 110 Mbytes hard-disk space
- One 3.5" High-Density floppy disk drive or ISO 9660 compatible CD-ROM drive
- VGA display
- Mouse
- One parallel and two serial ports
- 8 Mbytes of RAM for devices up to XC4006
- 16 Mbytes of RAM for XC4008, and above

Package Features - Viewlogic PC

Feature	VL Base	VL Std.	VLS Base	VLS Std.	VLS Ext.
Libraries and Interface	✓	✓	✓	✓	✓
Schematic Editor			✓	✓	✓
Simulator (Limited)			✓		
Simulator (Unlimited)				✓	✓
EPLD Devices	✓	✓	✓	✓	✓
FPGA Limited (1)	✓		✓		
FPGA 2K, 3K, 4K		✓		✓	✓
Core Implementation	✓(2)	✓	✓(2)	✓	✓
Synthesis Tools					✓
X-BLOX		✓		✓	✓
Parallel Download	✓				
XChecker Cable		✓	✓	✓	✓
FPGA Demo Board	✓	✓	✓	✓	✓
Hotline Support	✓	✓	✓	✓	✓
1 Yr Software Update		✓	✓	✓	✓

Note: VL = Viewlogic, VLS = Viewlogic *Stand-Alone*

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003

(2) XDE Design Editor not included

X5240

Viewlogic *Stand-alone* – Extended System (PC)

Extended *Stand-alone* System Includes:

- ViewDraw Schematic editor with library support for XC2000, XC3000/XC3100, XC4000 FPGAs and XC7200 and XC7300 EPLDs
- ViewSim Functional, Timing , and VHDL Simulation (unlimited gates)
- ViewSynthesis – VHDL synthesis with X-BLOX integration and library synthesis support for XC2000, XC3000/XC3100, XC4000 FPGAs
- X-BLOX Module Generator + Optimizer
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC2000, XC3000/XC3100, and XC4000
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for the first year

Note:

- The Viewlogic software (ViewDraw, ViewSim) included in this package is the DOS-based Workview 4.1 software

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

- Fully compatible PC386/486
- MS-DOS version 5.0 (minimum)
- Minimum 90 to 100 Mbytes hard-disk space
- One 3.5" High-Density floppy disk drive or ISO 9660 compatible CD-ROM drive
- VGA display
- Mouse
- One parallel and two serial ports
- 8 Mbytes of RAM for devices up to XC4006
- 16 Mbytes of RAM for XC4008 and above

Package Features - Viewlogic PC

Feature	VL Base	VL Std.	VLS Base	VLS Std.	VLS Ext.
Libraries and Interface	✓	✓	✓	✓	✓
Schematic Editor			✓	✓	✓
Simulator (Limited)			✓		
Simulator (Unlimited)				✓	✓
EPLD Devices	✓	✓	✓	✓	✓
FPGA Limited (1)	✓		✓		
FPGA 2K, 3K, 4K		✓		✓	✓
Core Implementation	✓(2)	✓	✓(2)	✓	✓
Synthesis Tools					✓
X-BLOX		✓		✓	✓
Parallel Download	✓				
XChecker Cable		✓	✓	✓	✓
FPGA Demo Board	✓	✓	✓	✓	✓
Hotline Support	✓	✓	✓	✓	✓
1 Yr Software Update		✓	✓	✓	✓

Note: VL = Viewlogic, VLS = Viewlogic *Stand-Alone*

(1) XC2000, XC3000, up to XC3042/XC3142; XC4000 up to XC4003

(2) XDE Design Editor not included

X5241

Viewlogic – Standard System (Sun-4)

Standard System Includes:

- Schematic Interface for ViewDraw with library support for XC2000, XC7200/XC7300 EPLDs and XC3000/XC3100 and XC4000 FPGAs
- Functional and Timing Simulation Interface for ViewSim
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC2000, XC3000/XC3100, and XC4000
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for the first year

Notes:

- This package does not include ViewDraw schematic capture or ViewSim simulation tools. They must be purchased separately.
- Interface supports Workview 4.1 and Powerview 5

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

- Sun OS 4.1.x
- Graphical monitor (color recommended)
- X-Windows or Open Windows support
- 32 Mbytes of RAM is highly recommended for XC3090, XC3190, XC3195 or XC4000 designs
- Swap Space: 50 Mbytes
- TCP/IP Software
- Minimum 80 to 100 Mbytes hard-disk space for Xilinx software
- CD-ROM Drive

Package Features - Viewlogic W/S

Feature	Std.
Libraries and Interface	✓
Schematic Editor	
Simulator (Limited)	
Simulator (Unlimited)	
EPLD Devices	✓
FPGA Limited	
FPGA 2K, 3K, 4K	✓
Core Implementation	✓
Synthesis Tools	
X-BLOX	✓
Parallel Download	
XChecker Cable	✓
FPGA Demo Board	✓
Hotline Support	✓
1 Yr Software Update	✓

X5243

Viewlogic – Standard System (HP700)

Standard System Includes:

- Schematic Interface for ViewDraw with library support for XC7200/XC7300 EPLDs and XC2000, XC3000/XC3100 and XC4000 FPGAs
- Functional and Timing Simulation Interface for ViewSim
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC2000, XC3000/XC3100 and XC4000
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for the first year

Notes:

- This package does not include ViewDraw schematic capture or ViewSim simulation tools. They must be purchased separately.
- Interface supports Workview 4.1 and Powerview 5

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

HP700 Series

- HPUX 9.0/9.01
- 50 to 150 Mbytes of RAM hard-disk space allocated for Xilinx designs
- 32 Mbytes of RAM (minimum)
- Color Monitor
- X11 R5 Windows Support
- HP_VUE 3.0
- Swap Space: 140 Mbytes (minimum)
- TCP/IP Software
- CD-ROM Drive

Recommended Hardware Environment:

- Additional RAM to increase performance

Package Features - Viewlogic W/S

Feature	Std.
Libraries and Interface	✓
Schematic Editor	
Simulator (Limited)	
Simulator (Unlimited)	
EPLD Devices	✓
FPGA Limited	
FPGA 2K, 3K, 4K	✓
Core Implementation	✓
Synthesis Tools	
X-BLOX	✓
Parallel Download	
XChecker Cable	✓
FPGA Demo Board	✓
Hotline Support	✓
1 Yr Software Update	✓

X5245

Mentor V8 – Standard System (Sun-4)

Standard System Includes:

- Mentor V8 Interface (Mentor Design Architect/ QuickSim II Libraries and Interface)
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC2000, XC3000/XC3100, and XC4000
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for the first year

Notes:

- This package does not include Design Architect schematic capture, or QuickSim II simulation tools. Contact your local Mentor Graphics sales office to purchase these tools.
- AutoLogic synthesis program, libraries and interface are available from Mentor Graphics.

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

SparcStation Series

- Sun OS 4.1X
- Mentor Graphics Version 8.2_5
- 50 to 200 Mbytes hard-disk space allocated for Xilinx designs
- 32 Mbytes of RAM (minimum)
- Color Monitor
- X11 R4 Windows Support
- Open Windows 3.0
- Swap Space: 125 Mbytes (minimum)
- TCP/IP Software
- CD-ROM Drive

Recommended Hardware Environment:

- Additional RAM to increase performance

Package Features - Mentor W/S

Feature	Std.
Libraries and Interface	✓
Schematic Editor	
Simulator (Limited)	
Simulator (Unlimited)	
EPLD Devices	✓(1)
FPGA Limited	
FPGA 2K, 3K, 4K	✓
Core Implementation	✓
Synthesis Tools	
X-BLOX	✓
Parallel Download	
XChecker Cable	✓
FPGA Demo Board	✓
Hotline Support	✓
1 Yr Software Update	✓

(1) Except Apollo Platform

X5246

Mentor V8 – Standard System (HP700 Series)

Standard System Includes:

- Mentor V8 Interface (Mentor Design Architect/ QuickSim II Libraries and Interface)
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC2000, XC3000/XC3100, and XC4000
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for the first year

Notes:

- This package does not include Design Architect schematic capture, or QuickSim II simulation tools. Contact your local Mentor Graphics sales office to purchase these tools.
- AutoLogic synthesis program, libraries and interface are available from Mentor Graphics.

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

HP700 Series

- HPUX 9.0/9.01
- Mentor Graphics Version 8.2_5
- 50 to 150 Mbytes of hard-disk space allocated for Xilinx designs
- 32 Mbytes of RAM (minimum)
- Color Monitor
- X11 R5 Windows Support
- HP_VUE 3.0
- Swap Space: 140 Mbytes (minimum)
- TCP/IP Software
- CD-ROM Drive

Recommended Hardware Environment:

- Additional RAM to increase performance

Package Features - Mentor W/S

Feature	Std.
Libraries and Interface	✓
Schematic Editor	
Simulator (Limited)	
Simulator (Unlimited)	
EPLD Devices	✓(1)
FPGA Limited	
FPGA 2K, 3K, 4K	✓
Core Implementation	✓
Synthesis Tools	
X-BLOX	✓
Parallel Download	
XChecker Cable	✓
FPGA Demo Board	✓
Hotline Support	✓
1 Yr Software Update	✓

(1) Except Apollo Platform

X5246

Mentor V8 – Standard System (HP400/Apollo)

V8 on the HP400/Apollo Platform is Not a Recommended Configuration

Standard System Includes:

- Mentor V8 Interface (Mentor Design Architect/ QuickSim II Libraries and Interface)
- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC7000, XC2000, XC3000/XC3100, and XC4000
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Software Support and Updates for the first year

Notes:

- This package does not include Design Architect schematic capture, or QuickSim II simulation tools. Contact your local Mentor Graphics sales office to purchase these tools.
- AutoLogic synthesis program, libraries and interface are available from Mentor Graphics.

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

Apollo DN 4xxx/5xxx or HP400 Series

- Apollo Domain Operating System SR10.4
- Mentor Graphics Version 8.2_5
- 50 to 200 Mbytes hard-disk space allocated for Xilinx designs
- 32 Mbytes of RAM (minimum)
- Color Monitor
- X11R4
- Display Manager
- Swap Space: 125 Mbytes (minimum)

Recommended Hardware Environment:

- Additional RAM to increase performance

Package Features - Mentor W/S

Feature	Std.
Libraries and Interface	✓
Schematic Editor	
Simulator (Limited)	
Simulator (Unlimited)	
EPLD Devices	✓(1)
FPGA Limited	
FPGA 2K, 3K, 4K	✓
Core Implementation	✓
Synthesis Tools	
X-BLOX	✓
Parallel Download	
XChecker Cable	✓
FPGA Demo Board	✓
Hotline Support	✓
1 Yr Software Update	✓

(1) Except Apollo Platform

X5246

Synopsys – Standard System (Sun-4)

Standard System Includes:

- XC3000/XC3100 and XC4000 synthesis library (DS-401)
- Support for Synopsys Design Compiler and FPGA Compiler
- Core implementation software for FPGAs (DS-502) with device support for XC3000/XC3100, and XC4000 families
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Translator from Synopsys to Xilinx XNF
- Design methodology notes for Synopsys
- Software Support and Updates for the first year

Notes:

- This package does not include Synopsys Design Compiler or FPGA Compiler. These must be purchased separately from Synopsys.
- Call Xilinx for availability of Synopsys VSS support

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

SparcStation Series

- Sun OS V4.1X
- Minimum 110 Mbytes hard-disk space for Xilinx software for Standard package (50 Mbytes for Interface only)
- 32 Mbytes of RAM
- Color Monitor
- X-Windows (R3 or R4)
- Openlook or Motif
- CD-ROM Drive

Package Features - Synopsys W/S

Feature	Std.
Libraries and Interface	✓
Schematic Editor	
Simulator (Limited)	
Simulator (Unlimited)	
EPLD Devices	
FPGA Limited	
FPGA 2K, 3K, 4K	✓
Core Implementation	✓
Synthesis Tools	
X-BLOX	✓
Parallel Download	
XChecker Cable	✓
FPGA Demo Board	✓
Hotline Support	✓
1 Yr Software Update	✓

X5247

Synopsys – Standard System (HP700 Series)

Standard System Includes:

- XC3000/XC3100 and XC4000 synthesis library (DS-401)
- Support for Synopsys Design Compiler and FPGA Compiler
- Core implementation software for FPGAs (DS-502) with device support for XC3000/XC3100, and XC4000 families
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Translator from Synopsys to Xilinx XNF
- Design methodology notes for Synopsys
- Software Support and Updates for the first year

Notes:

- This package does not include Synopsys Design Compiler or FPGA Compiler. These must be purchased separately from Synopsys.
- Call Xilinx for availability of Synopsys VSS support

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

HP700 Series

- HPUX 9.0/9.01
- Minimum 110 Mbytes hard-disk space for Xilinx software for Standard package (50 Mbytes for Interface only)
- 32 Mbytes of RAM
- Color Monitor
- Swap Space; 100 Mbytes (minimum)
- CD-ROM Drive

Package Features - Synopsys W/S

Feature	Std.
Libraries and Interface	✓
Schematic Editor	
Simulator (Limited)	
Simulator (Unlimited)	
EPLD Devices	
FPGA Limited	
FPGA 2K, 3K, 4K	✓
Core Implementation	✓
Synthesis Tools	
X-BLOX	✓
Parallel Download	
XChecker Cable	✓
FPGA Demo Board	✓
Hotline Support	✓
1 Yr Software Update	✓

X5247

Synopsys – Standard System (HP400/Apollo)

Standard System Includes:

- XC3000/XC3100 and XC4000 synthesis library (DS-401)
- Support for Synopsys Design Compiler and FPGA Compiler
- Core implementation software for FPGAs (DS-502) with device support for XC3000/XC3100, and XC4000 families
- X-BLOX Module Generator + Optimizer
- FPGA Demonstration Board
- XChecker Diagnostic Cable
- Translator from Synopsys to Xilinx XNF
- Design methodology notes for Synopsys
- Software Support and Updates for the first year

Notes:

- This package does not include Synopsys Design Compiler or FPGA Compiler. These must be purchased separately from Synopsys.
- Call Xilinx for availability of Synopsys VSS support.

Support and Updates Include:

- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX
- Software Updates for one year
- Documentation Updates

Required Hardware Environment:

Apollo DN 4xxx/5xxx or HP400 Series

- Apollo Operating System SR10.4
- Minimum 110 Mbytes hard-disk space for Xilinx software for STANDARD package (50 Mbytes for Interface only)
- 32 Mbytes of RAM
- Color Monitor
- XApollo
- Display Manager or Motif

Package Features - Synopsys W/S

Feature	Std.
Libraries and Interface	✓
Schematic Editor	
Simulator (Limited)	
Simulator (Unlimited)	
EPLD Devices	
FPGA Limited	
FPGA 2K, 3K, 4K	✓
Core Implementation	✓
Synthesis Tools	
X-BLOX	✓
Parallel Download	
XChecker Cable	✓
FPGA Demo Board	✓
Hotline Support	✓
1 Yr Software Update	✓

X5247

Cadence System (Sun-4 and HP700)

Description	Cadence Order Number	Contact Cadence to order these Products
• Xilinx Front End – Composer	51022	
• Xilinx Front End – Concept	51020	
• Verilog Interface	51022UG2	
• Xilinx Front to Back – Composer	51023	
• Xilinx Front to Back – Composer without X-BLOX	51023UG2	
• Xilinx Front to Back – Concept	51021	
• Xilinx Front to Back – Concept without X-BLOX	51021UG2	

Xilinx Front End Includes:

- Cadence Interface (Composer or Concept schematic symbols, Verilog or RapidSim simulation models and interfaces)
- X-BLOX Module Generator + Optimizer
- First year annual support contract

Xilinx Front-to-Back Includes:

All of the above plus

- Core implementation software for EPLDs (DS-550) and FPGAs (DS-502) with device support for XC2000, XC3000/XC3100, and XC4000
- FPGA Demonstration Board
- XChecker Diagnostic Cable

Note:

- All products listed above are available from Cadence

Required Hardware Environment:

SparcStation Series

- Sun OS 4.1X
- 60 Mbytes hard-disk allocated for Xilinx designs
- 32 Mbytes of RAM
- Color Monitor
- X-Windows (R3 or R4)
- Open Windows or Motif

Required Hardware Environment:

HP700 Series

- HPUX 9.0/9.01
- 50 to 150 Mbytes hard-disk space allocated for Xilinx designs
- 32 Mbytes of RAM (minimum)
- Color Monitor
- X11 R5 Windows Support
- HP_VUE 3.0
- Swap Space: 140 Mbytes (minimum)
- TCP/IP Software

Package Features - Cadence W/S

Feature	Front End	Front-to-Back	Verilog
Libraries and Interface	✓	✓	✓
Schematic Editor			
Simulator (Limited)			
Simulator (Unlimited)			
EPLD Devices			
FPGA Limited			
FPGA 2K, 3K, 4K	✓	✓	✓
Core Implementation		✓	
Synthesis Tools			
X-BLOX	✓	✓	
Parallel Download			
XChecker Cable		✓	
FPGA Demo Board		✓	
Hotline Support	✓	✓	✓
1 Yr Software Update	✓	✓	✓

X5248

Individual Product Descriptions

Table of Contents

FPGA Core Implementation – DS-502

Core Implementation Includes:

- Software to process an XNF file for an XC2000, XC3000/XC3100 or XC4000 device into a BIT or PROM file that can be downloaded
- Timing-driven router
- Incremental design
- Logic reduction algorithm
- Design rule checking
- Mapping software
- Automatic placement and routing
- Interactive placement and routing
- Static timing analysis
- Bitstream generation
- PROM file formatting
- XChecker diagnostic cable
- Support and updates for first year

Support and Updates Include:

- Software updates for one year
- Documentation updates
- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX

Hardware Requirements:

- Minimum 65 Mbytes hard-disk space for Xilinx software
- Other hardware requirements are same as the Standard package

XEPLD Translator for EPLDs – DS-550

The XEPLD Translator provides a complete, user-friendly, multi-platform design environment for creating behavioral or schematic designs. XEPLD allows users to easily create, verify, and implement logic designs targeting the entire range of Xilinx XC7000-series devices.

PAL Conversion

XEPLD provides an efficient tool for converting multiple PALs into a single high-performance EPLD. Designed to work with industry-standard PLD compilers and languages, XEPLD supports design entry and verification using the front end tools with which designers are already familiar. Then, acting as a device fitter, XEPLD quickly maps the design into a Xilinx EPLD. PAL equation files and behavioral equations can also be embedded in schematic designs.

Automatic Logic Mapping and Optimization

The automatic partitioning and mapping capabilities of XEPLD allow the designer to concentrate on design functionality without concern for physical implementation; all device resources are automatically mapped and interconnected with no user intervention required. In addition, automatic logic optimization insures the highest performance and the most efficient usage of device resources. Because of these automatic features, the user does not need a detailed knowledge of the device architecture. However, XEPLD also allows the designer to fully control the physical mapping of logic and I/O resources when necessary.

Feature Summary

- Behavioral Design Entry

In addition to the built-in PLUSASM Boolean equation assembler, XEPLD supports industry-standard PLD compilers such as ABEL, CUPL, and PALASM, providing a design environment with which users are already familiar.

- Schematic Design Entry

XEPLD provides a schematic library that includes familiar TTL and PAL components for use with industry-standard schematic editors such as those available from OrCAD, ViewLogic, Mentor Graphics, and Cadence Design Systems.

- Simulation Support

XEPLD supports various third-party simulators such as ViewLogic ViewSim, OrCAD VST, Mentor QuickSim, Cadence Verilog, and Cadence RapidSim. Both functional and timing simulation are supported.

- Board-Level Simulation Support

XEPLD device models are available from Logic Modeling Corporation for board-level simulation on a variety of platforms.

- High-Speed Compilation

Design iterations are easily performed and the results are quickly reported.

- Predictable Design Performance

The PAL-like architecture of the Xilinx EPLDs provides fixed predictable delays independent of physical placement, routing, or device utilization.

- Automatic Mapping and Logic Optimization

Device resources are automatically mapped for optimal efficiency and high performance. Users can focus on design functionality without concern for the physical implementation in the device.

- Complete Design Control

Users have the option to override the automatic features of XEPLD and selectively control any or all device resources.

- Multiple Platform Support

XEPLD runs on Sun, HP, and PC (DOS) platforms.

Schematic and Simulator Interfaces

Interfaces and libraries for several popular schematic editors and timing simulators are available as individual products, for users that already own an editor and simulator. For designers looking for a design entry tools, Xilinx offers Xilinx-specific versions of Viewlogic's ViewDraw schematic editor, ViewSim simulator, and ViewSynthesis VHDL synthesizer and VHDL simulator.

The following products are available for the platforms noted in parentheses:

DS-390 ViewDraw schematic editor with Xilinx libraries and interface (PC)

DS-290 ViewSim simulator with Xilinx libraries and interface (PC)

DS-391 Libraries and interfaces that support Viewlogic's Workview, PRO Series, Workview PLUS, and Powerview design entry and simulation tools (PC, Sun, HP700)

DS-344 Libraries and interfaces for Mentor Graphics V8 Design Architect schematic editor and QuickSim II simulator (HP700, Sun, Apollo/HP400)

DS-35 Libraries and interfaces for OrCAD 386+ schematic editor and VST 386+ simulator (PC)

Features

- Complete set of primitive and macro libraries for all FPGA and EPLD products
- Full simulation models provides for accurate post-layout timing analysis
- Unified libraries allow easy migration between all Xilinx architectures, including EPLDs
- Converts schematic drawings to Xilinx Netlist Format (XNF) output
- Converts XNF files to format compatible with logic and timing simulators
- Supports unlimited levels of hierarchy
- Includes one year of support and updates
- All above products can be purchased with core implementation tools as a package, offering easier upgrading and reduced cost.

X-BLOX – DS-380

X-BLOX Includes:

- Parameter-based schematic and function generation tool. Allows block-diagram design entry using generic function modules.
- Works with many Schematic Entry tools (Viewlogic, Mentor, OrCAD, Cadence and others)
- Expert system that automatically utilizes the advanced features of the XC4000 family and XC3000A family
- Schematic library with more than 30 frequently-used generic modules (adders, counters, decoders, registers, MUXes, etc.)
- First year support and updates

Support and Updates Include:

- Software updates for one yar
- Documentation updates
- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX

Note:

- Supports XC4000 and XC3000A/XC3100A families. Does not support XC2000, XC3000/XC3100 or XC7000 families.
- Base system requires additional five Mbytes hard-disk space for program and design files.

Xilinx ABEL Design Entry – DS-371

The Xilinx ABEL system gives designers the ability to enter Xilinx designs using the industry standard ABEL Hardware Description Language (ABEL-HDL). Designers can describe circuits with Boolean equations, state machines and truth tables. State machine and logic optimization software automatically generates efficient logic for Xilinx devices.

Many designs contain portions of logic that are best described in a text-based format; some designs can be completely described in this way. In the Xilinx ABEL system, Xilinx designs can be created with Boolean equations, state machines, and truth tables. The ABEL HDL makes designing quick and simple. Intelligent state machine and logic optimization software automatically creates efficient, fast state machines. The ABEL simulator allows functional simulation of ABEL-HDL designs.

EPLD designs may be entered entirely with ABEL-HDL. FPGA designs should be entered via a combination of XABEL and a schematic editor to take optimal advantage of the Xilinx architectures. The recommended design flow is to enter designs schematically with functional blocks that refer to logic described in ABEL-HDL. From inside the Xilinx ABEL environment, designers create and compile the logic in these functional blocks. The Xilinx XMake program then compiles the complete design to a bitstream that can be downloaded to a Xilinx device. XMake automatically calls the software that merges the various design files (schematics and ABEL-HDL), partitions, places and routes the design and creates the final bitstream. The design can then be verified with a simulator and a timing analyzer, as well as verified in-circuit.

One-Hot Encoding

For the flop-flop rich, fan-in limited Xilinx FPGA architecture, One-Hot Encoding (OHE) is the preferred technique for implementing high-performance state machines. OHE is also know as state-per-bit encoding since it uses one flip-flop per state. OHE takes advantage of the abundance of flip-flops in Xilinx FPGAs to reduce the levels of logic required to implement a state machine. This implementation significantly increases performance over fully encoded state machines, the traditional technique used in PLDs. Xilinx ABEL automatically uses OHE on symbolic state machines created in ABEL-HDL for FPGAs.

Features

- State Machine and Boolean equation entry via Data I/O ABEL language
- ABEL Functional Simulator
- Xilinx-specific ABEL environment, compiler, and optimizer for FPGAs (XC2000, XC3000, XC4000) and EPLDs (XC7000)
- Automatic symbolic One-Hot Encoding or fully encoded state-machine implementation
- Ability to integrate ABEL designs with other schematic elements

Support and Updates

- Software updates for one year
- Documentation updates
- Hotline Telephone Support
- Access to Xilinx bulletin board
- Apps FAX

Additional Hardware Requirements

- 5 Mbytes hard-disk space for program and design files

Xilinx-Synopsys Interface (XSI) – DS-401

This interface and library product supports VHDL and Verilog/HDL synthesis using either the Synopsys Design Compiler or FPGA Compiler products

Features

- XC3000/XC3100 and XC4000 synthesis library
- X-BLOX synthetic library
- Translator from Synopsys to Xilinx XNF
- Ability to integrate models with other design
- Available for Sun-4, HP700, and HP400/Apollo platforms

DS-401 (XSI) lets the Synopsys FPGA Compiler and Design Compiler target the XC3000, XC3100, and XC4000 FPGA families. XSI consists of synthesis libraries, a translator from Synopsys to XNF, and a library of X-BLOX functions implemented using Synopsys DesignWare.

Language Support

Either VHDL or Verilog/HDL entry is supported through the use of the appropriate Synopsys language compiler.

Compiler Support

FPGA Compiler is *highly* recommended for XC4000 designs due to its specific XC4000 algorithms. Design Compiler is sufficient for XC3000 and XC3100 designs.

Simulation Support

Behavioral simulation before compilation using Synopsys VHDL System Simulator (VSS) is supported. In the future, gate-level simulation of designs after layout will be supported as well.

Support and Updates

- Software updates for one year
- Documentation updates
- Hotline Telephone Support for the first six months
- Access to Xilinx bulletin board
- Apps FAX

Notes

- This product does not support the Synopsys Test Compiler
- A Synopsys Standard package is available which combines XSI (DS-401) and FPGA core implementation tools (DS-502) in one product. Packages offer reduced prices over modules purchased separately.
- The X-BLOX library allows Synopsys software to automatically insert certain X-BLOX functions (adders, subtractors, and comparators) where possible for maximum performance. In-warranty XSI customer receive X-BLOX as an automatic upgrade.

Parallel Download and XChecker Cables

A download or XChecker cable is included in each of the bundled packages, and in the DS-502 Core Implementation product. Additional cables can be ordered separately.

Parallel Download Cable Features

- Provides bitstream and PROM file download capability
- Works with parallel ports on IBM '386/'486 and compatibles
- Compatible with XChecker diagnostics software and the XACT Probe utility
- Flying wire and flat header jumpers provide easy access during prototyping
- Order code: HW-DWNCBL-PC1

The Parallel Download Cable package includes the download cable with 25-pin connector, flying-wire jumper, and flat-head jumper.

XChecker Cable Features

- Provides bitstream and PROM file download capability
- Provides readback capability
- Works with serial ports on IBM '386/'486 and compatibles
- Works with serial ports on Sun and HP/Apollo workstations
- Compatible with XChecker diagnostics software and the XACT Probe utility
- Flying wire and flat header jumpers provide easy access during prototyping
- Order Code: HW-XCHCBL-PC1 (or SN2, or HP7, or AP1)

The XChecker Cable package includes the X-Checker diagnostics test fixture, the XChecker cable, flying-wire jumper, and flat-head jumper.

3 Volt XChecker Adapter (optional)

- Connects to XChecker cable
- On board DC to DC converter
- Level translation for logic signals
- Accepts any V_{CC} supply voltage from 2.9 V to 5.2 V
- Order Code: HW-XCH3V

Demonstration Boards

There are three different Xilinx Demonstration Boards, two dedicated to the XC3000 and XC4000 families, respectively, and a new general purpose FPGA Demo Board, which is now included in the software packages, and can also be ordered separately.

All demo boards are compatible with XChecker and the Download cable, and all require 5-V power source.

XC3000/XC3100 Demo Board Features

- XC3020 in 68-pin PLCC socket
- One 7-segment display
- One 8-segment bar display
- Eight DIP switches
- Momentary contact switches for Program and Reset
- Supports Master Serial configuration mode with interface to XC17000 serial PROM
- Socket can accomodate any XC3000/XC3100 device in PC68

XC4000 Demo Board Features

- XC4003 in 84-pin PLCC socket
- Two 7-segment displays
- One 8-segment bar display
- Eight DIP switches for inputs to LCA devices
- Momentary contact switches for Program, Reset, and Spare
- Supports Master Serial configuration mode with interface and sockets for up to three daisy-chained XC17000 serial PROMs
- Socket can accomodate any XC3000/XC3100 device in PC68

FPGA Demo Board Features

- XC4003 in 84-pin PLCC socket and XC3020 in 68-pin PLCC socket
- Three 7-segment displays, one for XC3000, two for XC4000
- Two 8-segment bar displays, one for XC3000, one for XC4000
- Two sets of eight DIP switches, one for XC3000, one for XC4000
- Test pins for access to all I/Os
- Momentary contact switches for Program, Reset, and Spare
- Supports Master Serial configuration mode with interface and sockets for up to three daisy-chained XC17000 serial PROMs (XC4000 as master)
- Order code: HW-FPGABOARD

XAPP 000.002 💾

General

💾 This application directory is available on the Xilinx Technical Bulletin Board as XAPP000. Design files of other application notes designated with a disk symbol are available under their own XAPP number.

Counters

Counter Performance Summary

	Loadable	Up	Down	Up/ Down	8-Bit		10-Bit		12-Bit		16-Bit		20-Bit		24-Bit		32-Bit	
					MHz	CLBs	MHz	CLBs	MHz	CLBs	MHz	CLBs	MHz	CLBs	MHz	CLBs	MHz	CLBs
XC3100-3																		
XAPP 001		•			173	5	116	8	108	9	107	14	103	17	103	21		
XAPP 002	•	•	•	•	47	8	38	10	37	12	29	16	22	20	22	24		
XAAP 002	•	•	•	•							41	17						
XAPP 003	•	•	•		63	9			52	15	48	20						
XAPP 004		•	•								54	23					37	49
XAPP 004				•							46	27					37	56
XAPP 014		•									204*	24						
XC3000-125																		
XAPP 001		•			81	5	60	8	56	9	57	14	55	17	55	21		
XAPP 002	•	•	•	•	26	8	21	10	21	12	17	16	13	20	11	24		
XAPP 002	•	•	•	•							24	17						
XAPP 003	•	•	•		33	9			29	15	26	20						
XAPP 004		•	•								30	23					21	49
XAPP 004				•							25	27					20	56
XAPP 014		•									95*	24						
XC4000-5																		
XAPP 014		•									111*	17						

* Estimated

X3200

Arithmetic Functions

Adders, Subtracters and Accumulators in XC3000—XAPP 022.000 **8-98**
This Application Note surveys the different adder techniques that are available for XC3000 designs. Examples are shown, and a speed/size comparison is made.

Using the Dedicated Carry Logic in XC4000—XAPP 013.001 **8-105**
This Application Note describes the operation of the XC4000 dedicated carry logic, the standard configurations provided for its use, and how these are combined into arithmetic functions and counters.

Estimating the Performance of XC4000 Adders and Counters—XAPP 018.000 **8-116**
Using the XC4000 dedicated carry logic, the performance of adders and counters can easily be predicted. This Application Note provides formulae for estimating the performance of such adders and counters.

Calculating XC7200 Arithmetic Performance—XAPP 032.001 **8-119**
This Application Note describes how to estimate the performance of arithmetic circuits that are implemented using the XC7200 dedicated carry circuitry.

18-Bit Pipelined Accumulator—XAPP 039.001 **8-121**
This Application Note describes a pipelining technique that significantly improves the throughput of an accumulator.

Special Purpose Memory

Miscellaneous Applications

Additional XC4000 Data

XAPP 045.000

Application Note By PHILIP FREIDIN

Summary

This Application Note contains additional information that may be of use when designing with the XC4000 families of devices. This information supplements the product descriptions and specifications, and is provided for guidance only.

Xilinx Family

XC4000/XC4000A/XC4000H

Introduction

This application note describes the electrical characteristics of the output drivers, their static output characteristics or I/V curves, the additional delay caused by capacitive loading, and the ground bounce created when many outputs switch simultaneously.

Voltage/Current Characteristics of XC4000-Family Outputs

Figures 1 through 4 show the output source and sink currents, both drawn as absolute values. Note that the XC4000 families have an n-channel only, totem-pole like output structure that pulls a High output to a voltage level that is one threshold drop lower than V_{CC}. When driving inputs that have a 1.4-V threshold, this lower V_{OH} offers faster speed and more symmetrical switching delays. The XC4000H outputs offer an optional p-channel output driver and thus rail-to-rail switching a configuration option for each individual pin.

These curves represent typical devices. Measurements were taken at V_{CC}= 5 V, T = 25°C. These characteristics vary by manufacturing lot, and will be affected by future changes in minimum device geometries, notably a change from 0.8 µ to 0.6 µ. These characteristics are not production-tested as part of the normal device test procedure; they can, therefore, not be guaranteed. Although these measurements show that the output sink and source capability far exceeds the guaranteed data sheet limits, continuous high-current operation beyond the data sheet limits can cause metal migration of the on-chip metal traces, permanently damaging the device. Output currents in excess of the data-sheet limits are, therefore, not recommended for continuous operation. These output characteristics can, however, be used to calculate or model output transient behavior.

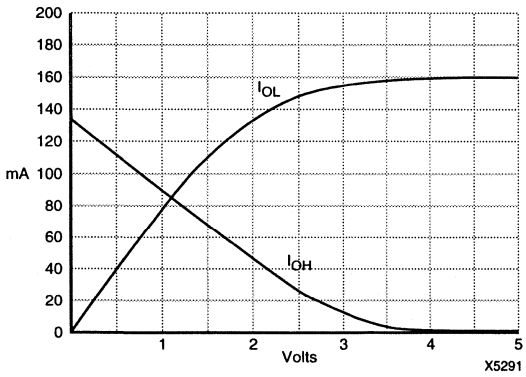


Figure 1. Output Voltage/Current Characteristics for XC4005-5

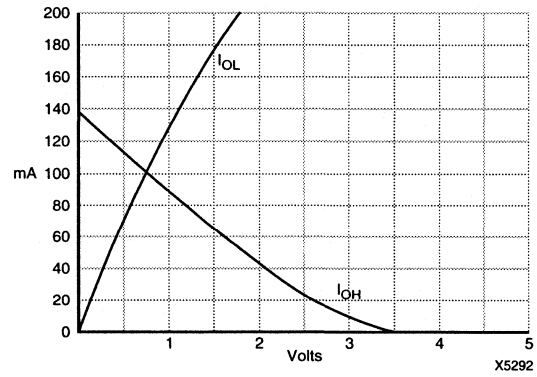


Figure 2. Output Voltage/Current Characteristics for XC4002A

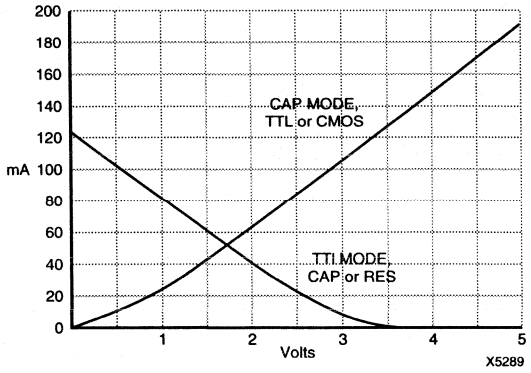


Figure 3. Output Voltage/Current Characteristics for XC4005H

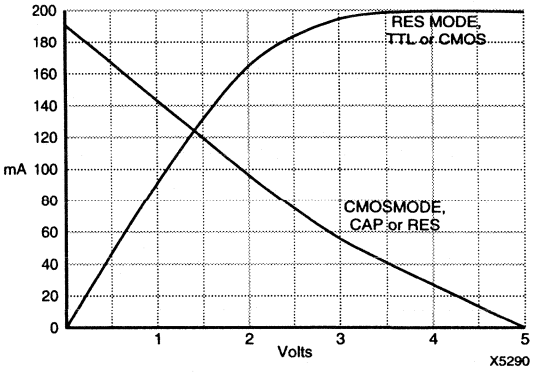


Figure 4. Output Voltage/Current Characteristics for XC4005H

Additional Output Delays When Driving Capacitive Load

Xilinx Product Specifications in Section 2 give guaranteed worst-case output delays with a 50-pF load.

The values given in Table 1 are actual measurements on a small number of mid-93 production XC4005-5, XC4005A-5 and XC4003H-5 devices, all in PQ208 packages, measured at room temperature and V_{CC} = 5.5 V. Listed is the additional output delay, measured crossing 1.5 V, relative to the delays specified in this 1994 Data Book, Section 2.

These parameters are not part of the normal production test flow, and can, therefore, not be guaranteed.

Table 2 lists the additional output delay, measured crossing 1.5 V, relative to the delay with 100 pF load shown in Table 1.

Example:

ΔT High-to-Low for XC4005-5 with Fast-mode output driving 250 pF:

1.2 ns (from Table 1) plus (250-100) pF • 1.5 ns/100 pF = 1.2 ns + 2.25 ns = 3.45 ns

Total propagation delay, clock OK to pad:

T_{OKPOF} + 3.45 ns = 7.0 ns + 3.45 ns = 10.45 ns

Table 1. Increase in Output Delay When Driving Light Capacitive Loads (<150 pF)

Family	Slew Mode	High-to-Low 10	50	100	Low-to-High 10	50	100	pF
XC4000	Slow	-1.6	0*	1.4	-1.4	0*	1.4	ns
	Fast	-1.6	0*	1.2	-1.2	0*	1.1	ns
XC4000A	Slow	-2.2	0*	1.7	-1.5	0*	1.4	ns
	MedSlow	-1.8	0*	1.6	-1.3	0*	1.1	ns
	MedFast	-1.8	0*	1.3	-1.3	0*	1.1	ns
	Fast	-2.0	0*	1.2	-1.0	0*	1.3	ns
XC4000H	Cap-CMOS	-2.2	0*	1.9	-0.5	0*	0.7	ns
	Res-CMOS	-1.4	0*	1.2	-1.0	0*	0.8	ns
	Cap-TTL	-1.9	0*	1.6	-1.2	0*	1.2	ns
	Res-TTL	-1.1	0*	1.0	-1.1	0*	1.0	ns

*Zero by definition.

Table 2. Increase in Output Delay When Driving Heavy Capacitive Loads (>150 pF)

Family	Slew Mode	High-to-Low	Low-to-High	
XC4000	Slow	1.7	1.2	ns/100 pF
	Fast	1.5	1.2	ns/100 pF
XC4000A	Slow	2.1	1.2	ns/100 pF
	MedSlow	1.5	1.1	ns/100 pF
	MedFast	1.0	1.1	ns/100 pF
	Fast	0.9	1.1	ns/100 pF
XC4000H	Cap-CMOS	2.7	0.9	ns/100 pF
	Res-CMOS	1.8	1.0	ns/100 pF
	Cap-TTL	2.1	1.3	ns/100 pF

Ground Bounce in XC4000 Devices

Ground-bounce is a problem with high-speed digital ICs, when multiple outputs change state simultaneously causing undesired transient behavior on an output, or in the internal logic. This is also referred to as the Simultaneous Switching Output (SSO) problem. Ground bounce is primarily due to current changes in the combined inductance of ground pins, bond wires, and ground metallization. The IC-internal ground level deviates from the external system ground level for a short duration (a few nanoseconds) after multiple outputs change state simultaneously. Ground bounce affects outputs that are supposed to be stable Low, and it also affects all inputs since they interpret the incoming level by referencing it to the internal ground. If the ground bounce amplitude exceeds the actual instantaneous noise margin, then a non-changing input will be interpreted as a short pulse with a polarity opposite to the ground bounce.

V_{CC} bounce is not as important as ground bounce, because it is of lower magnitude due to the weaker pull-up transistors. Also, the noise immunity in the High state is usually better than in the Low state, and input levels are referenced to ground, not V_{CC}. All this is the result of our industry's TTL heritage.

Test Method

Data was taken on XC40005-5, XC40005A-5, and XC40003H-5 devices, all in the PQ208 package, soldered to the Xilinx Ground Bounce Test Board. Pin 82, two pins away from the nearest ground pin, was configured as a permanently Low output driver, effectively monitoring the internal ground level. The simultaneously switching outputs were on pins 80 and 83, for two outputs switching; additionally, pins 80 and 86 were used for four outputs switching (81 and 84 on the XC40003H). The closest ground pins are 79 and 90.

Four ground-bounce parameters were measured at room temperature, with Vcc set at 5.5 V as shown in Figure 1.

V_{OLP-HL} Peak ground noise when outputs switch High-to-Low

V_{OLV-HL} Valley ground noise when outputs switch High-to-Low

V_{OLP-LH} Peak ground noise when outputs switch Low-to-High

V_{OLV-LH} Valley ground noise when outputs switch Low-to-High

All four parameters can affect system reliability.

The two positive peak values can cause problems with a signal leaving the ground bounce chip, driving another chip. The positive ground bounce voltage is added to the V_{OL}, and may exceed the receiving input's noise margin. A continuously logic Low input may thus be interpreted as a short-duration High pulse.

The two negative valley parameters can cause problems with a signal arriving at the ground-bounce chip, where the on-chip ground reference is negative, reducing the Low-level noise immunity. The incoming voltage may not be Low enough, and may, therefore, be interpreted as a short-duration High input pulse.

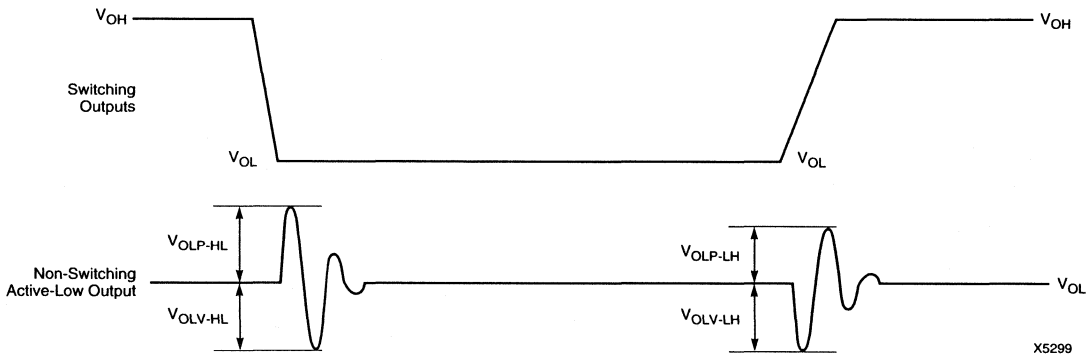


Figure 5. Ground Bounce

Table 3. Ground Bounce, 16 Outputs Switching, Each With 50 pF Load, V_{CC} = 5.5 V

Family	Slew Mode	High-to-Low		Low-to-High		
		V_{OLP}	V_{OLV}	V_{OLP}	V_{OLV}	
XC4000	Slow	670	480	240	240	mV
	Fast	1,170	710	480	660	mV
XC4000A	Slow	565	425	290	310	mV
	MedSlow	950	610	500	780	mV
	MedFast	1,140	860	500	780	mV
	Fast	1,240	910	500	810	mV
XC4000H	Cap-CMOS	940	660	660	770	mV
	Res-CMOS	1,250	1,210	590	480	mV
	Cap-TTL	830	460	450	570	mV
	Res-TTL	1060	980	440	350	mV

Table 4. Ground Bounce, 16 Outputs Switching, Each With 150 pF Load, V_{CC} = 5.5 V

Family	Slew Mode	High-to-Low		Low-to-High		
		V_{OLP}	V_{OLV}	V_{OLP}	V_{OLV}	
XC4000	Slow	740	330	210	280	mV
	Fast	1,180	420	350	710	mV
XC4000A	Slow	615	270	245	330	mV
	MedSlow	960	310	820	370	mV
	MedFast	1,140	620	370	790	mV
	Fast	1,200	640	370	810	mV
XC4000H	Cap-CMOS	1,080	390	470	860	mV
	Res-CMOS	1,500	820	420	590	mV
	Cap-TTL	900	250	320	610	mV
	Res-TTL	1,170	660	300	470	mV

Interpretation of the results

Ground bounce is a linear phenomenon. When multiple outputs switch, the total ground bounce is the sum of the ground-bounce values caused by individual outputs switching. Since the actual switching of multiple outputs is usually not quite simultaneous, small timing differences between the switching outputs, caused by routing delays, can indirectly affect the amplitude. With low capacitive loading, < 50 pF, the peaks and valleys might even partially cancel each other. With larger capacitive loads, the tendency is for valleys to combine with valleys and peaks to combine with peaks.

In most devices tested, the load capacitance does not directly affect the ground-bounce **amplitude**, but it does affect the **duration** of the ground-bounce signals.

On the fastest outputs, minimal load capacitance created a ground-bounce resonant frequency of 340 MHz, with a half-cycle time of 1.5 ns. Such a signal exceeds 90% of its peak amplitude for about 0.4 ns.

With a 50 pF load on the switching outputs, the ground bounce resonant frequency is 85 to 97 MHz, with a half-cycle time of 5 to 6 ns, staying 1.7 ns above 90% of peak amplitude.

With a 150 pF load on the switching outputs, the ground bounce resonant frequency is 40 to 60 MHz, with a half-cycle time of 8 to 12 ns, staying 3 ns above 90% of peak amplitude.

The main problem with large load capacitances is not an increase in amplitude, but rather an increase in duration of the ground-bounce signal. The amplitude is mainly affected by the number of outputs switching simultaneously, and by the slew-rate mode of these outputs. Switching outputs closer to the monitoring output also cause larger peaks and valleys than outputs further away.

Guidelines for reducing ground-bounce effects

- Minimize the impedance of the system ground distribution network and its connection to the IC pins. PQFPs are best suited, PGAs are worst, and PLCCs are in-between.
- Use PC-boards with ground- and V_{CC}-planes, connected directly to the ICs' supply pins. Place decoupling capacitors very close to these ground and V_{CC} pins.
- Keep the ground plane as undisturbed as possible. A row of vias can easily cause a dynamic ground-voltage drop.
- Keep the clock inputs physically away from the outputs that create ground bounce, and connect clocks to input pins that are close to a ground pin. Make sure that all clock and asynchronous inputs have ample noise margin, especially in the Low state.
- If possible, avoid simultaneous switching by staggering output delays, e.g. through additional local routing of signals or clocks.
- Spread simultaneously switching outputs around the IC periphery. For a 16-bit bus, use two outputs each on either side of four ground pins.

Ground-Bounce vs Delay Trade-Off

After the external sources of ground bounce have been reduced or eliminated. the designer can trade reduced ground bounce for additional delay by selecting between families and slew-rate options. Figures 2 and 3 show the available choices, based on 16 outputs switching simultaneously High-to-Low.

Summary

For light capacitive loads, the XC4000 and XC4000A, both in slow mode perform well, with ground bounce below 800 mV; the additional delay, compared to fast mode, is only 4 to 5 ns. For larger capacitive loads, the XC4000H in Capacitive-TTL mode offers the best trade-off.

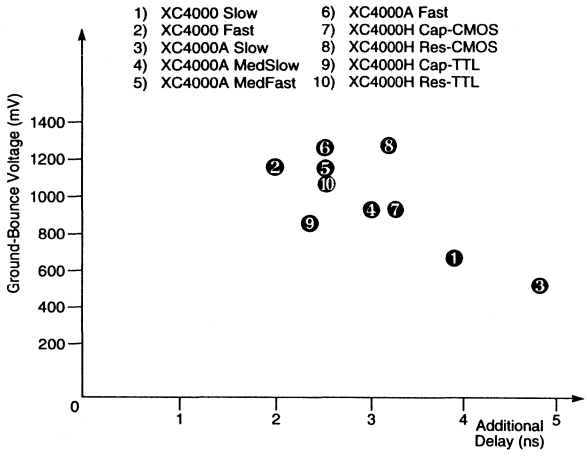


Figure 6. Ground-Bounce vs Delay Trade-off for 16 Outputs Switching 50 pF each

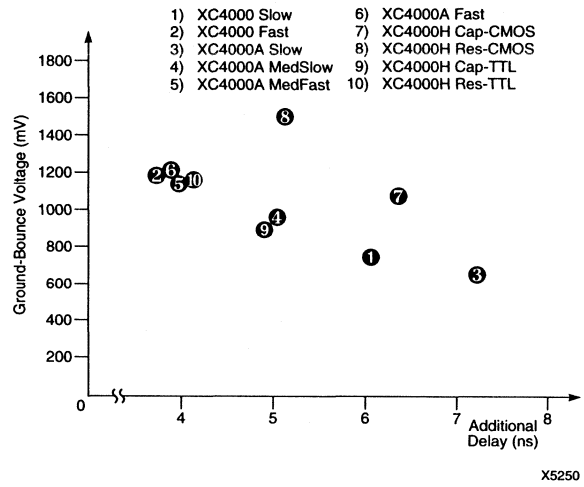


Figure 7. Ground-Bounce vs Delay Trade-off for 16 Outputs Switching 150 pF each

Additional XC3000 Data

XAPP 024.000

Application Note By PETER ALFKE and BERNIE NEW

Summary

This Application Note contains additional information that may be of use when designing with the XC3000 class of LCA devices. This information supplements the data sheets, and is provided for guidance only.

Xilinx Family

XC3000/XC3000A/XC3000L/XC3100/XC3100A

Introduction

The background information provided in this Application Note supplements the XC3000, XC3000A, XC3000L and XC3100 data sheets. It covers a wide range of topics, including a number of electrical parameters not specified in the data sheets, and unless otherwise noted, applies to all four families. These additional parameters are sufficiently accurate for most design purposes; unlike the parameters specified in the data sheets, however, they are not worst-case values over temperature and voltage, and are not 100% production tested. They can, therefore, not be guaranteed.

Configurable Logic Blocks

The XC3000/XC3100 CLB, shown in Figure 1, comprises a combinatorial function generator and two D-type flip-flops. Two output pins may be driven by either the

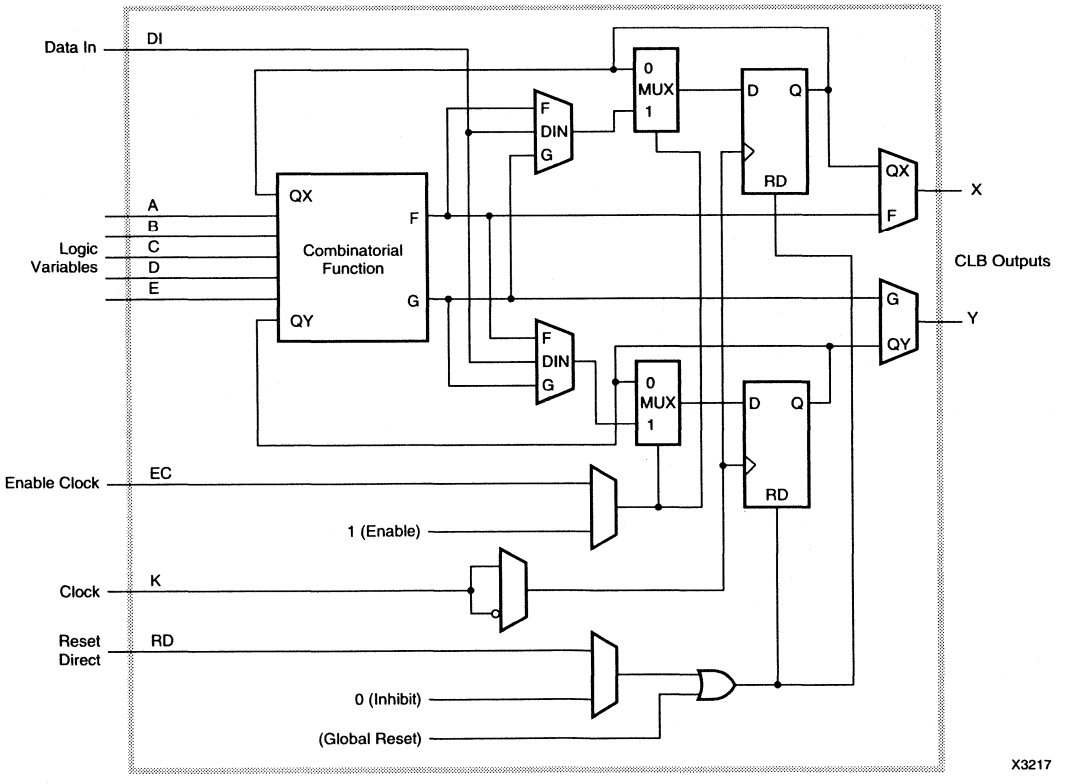


Figure 1. Configurable Logic Block (CLB)

function generators or the flip-flops. The flip-flop outputs may be routed directly back to the function generator inputs without going outside of the CLB.

The function generator consists of two 4-input look-up tables that may be used separately or combined into a single function. Figure 2 shows the three available options. Since the CLB only has five inputs to the function generator, inputs must be shared between the two look-up tables.

In the FG mode, the function generator provides any two 4-input functions of A, B and C plus D or E; the choice between D and E is made separately for each function. In the F mode, all five inputs are combined into a single 5-input function of A, B, C, D and E. Any 5-input function may be emulated. The FGM mode is a superset of the F mode, where two 4-input functions of A, B, C and D are multiplexed together according to the fifth variable, E.

In all modes, either of the B and C inputs may be selectively replaced by QX and QY, the flip-flop outputs. In the FG mode, this selection is made separately for the two look-up tables, extending the functionality to any two functions of four variable chosen from seven, provided two of the variables are stored in the flip-flops. This is particularly useful in state-machine-like applications.

In the F mode, the function generators implement a single function of five variables that may be chosen from seven, as described above. The selection of QX and QY is constrained to be the same for both look-up tables. The FGM mode differs from the F mode in that QX and QY may be selected separately for the two look-up tables, as in the FG mode. This added flexibility permits the emulation of selected functions that can include all seven possible inputs.

The automatic logic-partitioning software in the XACT development system only uses the FG and F modes. However, all three modes are available with manual partitioning, which may be performed in the schematic. If FG or F modes are required, it is simply a matter of including in the schematic CLBMAPs that define the inputs and outputs of the CLB.

The FGM mode is only slightly more complicated. Again, a CLBMAP must be used, with the signal that multiplexes between the two 4-input functions locked onto the E pin. The CLB will be configured in the FGM mode if the logic is drawn such that the gates forming the multiplexer are shown explicitly with no additional logic merged into them.

The two D-type flip-flops share a common clock, a common clock enable, and a common asynchronous reset signal. An asynchronous preset can be achieved using the asynchronous reset if data is stored in active-low form; the Low created by reset corresponds to the bit being asserted. The flip-flops cannot be used as latches.

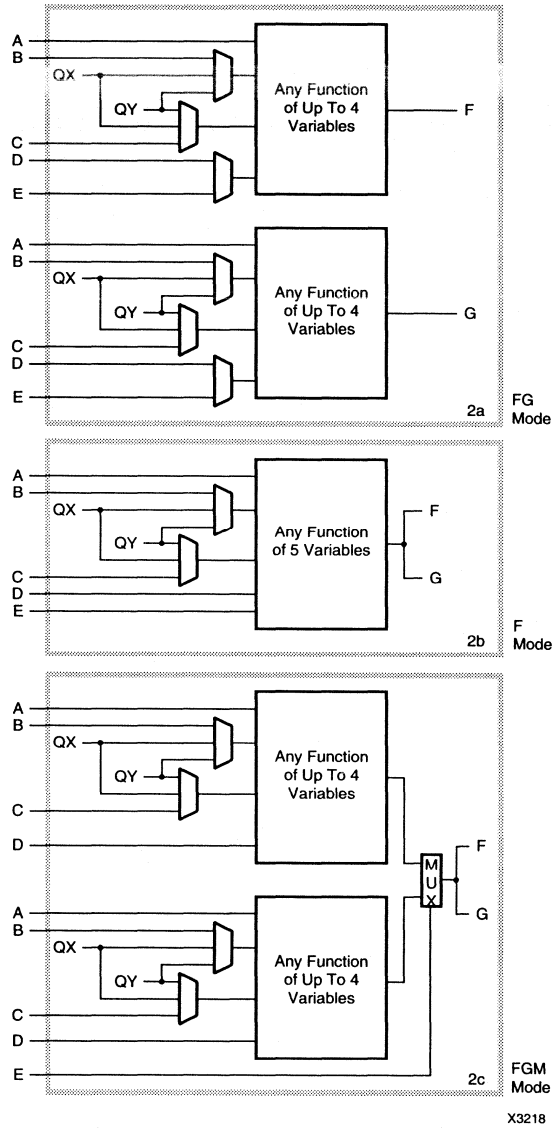


Figure 2. CLB Logic Options

If input data to a CLB flip-flop is derived directly from an input pad, without an intervening flip-flop, the data-pad-to-clock-pad hold time will typically be non-zero. This hold time is equal the delay from the clock pad to the CLB, but may be reduced according to the 70% rule, described later in the IOB Input section of this Application Note. Under this rule, the hold time is reduced by 70% of the delay from the data pad to the CLB, excluding the CLB

Table 1. Longline to CLB Direct Access

	CLB								TBUF
Longline	A	B	C	D	E	K	EC	RD	T
Left Most Vertical (GCLK)						X			
Left Middle Vertical		X					X	X	X
Right Middle Vertical			X		X				
Right Most Vertical (ACLK)						X			
Upper Horizontal				X					
Lower Horizontal	X							X	

set-up time. The minimum hold time is zero, even when applying the 70% rule results in a negative number.

The CLB pins to which Longlines have direct access are shown in Table 1. Note that the clock enable pin (EC) and the TBUF control pin are both driven from to the same vertical Long Line. Consequently, EC cannot easily be used to enable a register that must be 3-stated onto a bus. Similarly, EC cannot easily be used in a register that uses the Reset Direct pin (RD).

Input/Output Blocks

The XC3000/XC3100 IOB, shown in Figure 3, includes a 3-state output driver that may be driven directly or registered. The polarities of both the output data and the 3-state control are determined by configuration bits. Each output buffer may be configured to have either a fast or a slow slew rate.

The IOB input may also be direct or registered. Additionally, the input flip-flop may be configured as a latch. When an IOB is used exclusively as an input, an optional pull-up resistor is available, the value of which is 40-150 kΩ. This resistor cannot be used when the IOB is configured as an output or as a bidirectional pin.

Unused IOBs should be left unconfigured. They default to inputs pulled High with the internal resistor.

Inputs

All inputs have limited hysteresis, typically in excess of 200 mV for TTL input thresholds and in excess of 100 mV for CMOS thresholds. Exceptions to this are the $\overline{\text{PWRDWN}}$ pin, and the XTL2 pin when it is configured as the crystal oscillator input.

Experiments show that the input rise and fall times should not exceed 250 ns. This value was established through a worst-case test using internal ring oscillators to drive all I/O pins except two, thus generating a maximum of on-chip noise. One of the remaining I/O pins was configured as an input, and tested for single-edge response; the other I/O was used as an output to monitor the response.

These test conditions are, perhaps, overly demanding, although it was assumed that the PC board had negligible ground noise and good power-supply decoupling. While conservative, the resulting specification is, in most instances, easily satisfied.

IOB input flip-flops are guaranteed to operate correctly without data hold times (with respect to the device clock-input pad) provided that the dedicated CMOS clock input pad and the GCLK buffer are used. The use of a TTL clock or a different clock pad will result in a data-hold-time requirement. The length of this hold time is equal to the delay from the actual clock pad to the GCLK buffer minus the delay from the dedicated CMOS clock pad to the GCLK buffer.

To ensure that the input flip-flop has a zero hold time, delay is incorporated in the D input of the flip-flop, causing it to have a relatively long set-up time. However, the set-up time specified in the data sheet is with respect to the clock reaching the IOB. Since there is an unavoidable delay between the clock pad and the IOB, the input-pad-to-clock-pad set-up time is actually less than the data sheet number.

Part of the clock delay can be subtracted from the internal set-up time. Ideally, all of the clock delay could be subtracted, but it is possible for the clock delay to be less than its maximum while the internal set-up time is at its maximum value. Consequently, it is recommended that, in a worst-case design, only 70% of the clock delay is subtracted.

The clock delay can only be less than 70% of its maximum if the internal set-up time requirement is also less than its maximum. In this case, the pad-to-pad set-up time actually required will be less than that calculated.

For example, in the XC3000-125, the input set-up time with respect to the clock reaching the IOB is 16 ns. If the delay from the clock pad to the IOB is 6 ns, then 70% of this delay, 4.2 ns, can be subtracted to arrive at a maximum pad-to-pad set-up time of ~12 ns.

The 70% rule must be applied whenever one delay is subtracted from another. However, it is recommended that delay compensation only be used routinely in connection with input hold times. *Delay compensation in asynchronous circuits is specifically not recommended.* In any case, the compensated delay must not become negative. If 70% of the compensating delay is greater than the delay from which it is deducted, the resulting delay is zero.

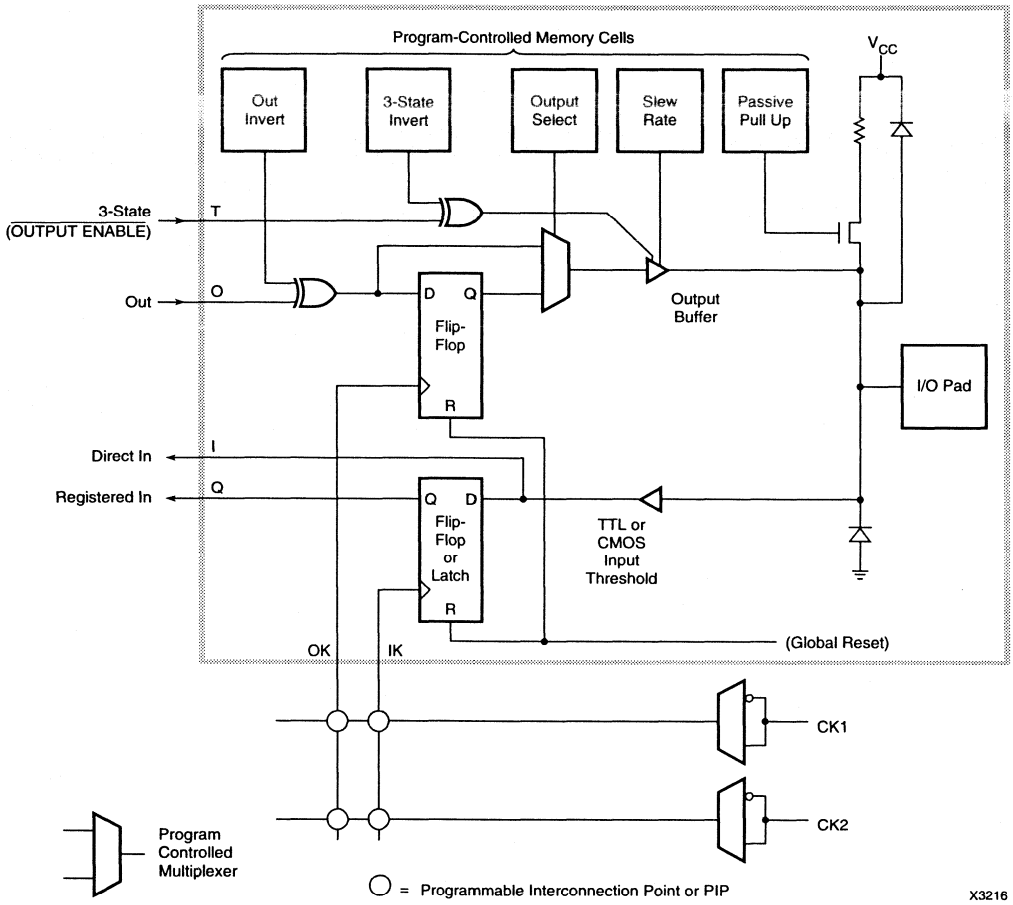


Figure 3. Input/Output Block (IOB)

The 70% rule in no way defines the absolute minimum values delays that might be encountered from chip to chip, and with temperature and power-supply variations. It simply indicates the relative variations that might be found within a specific chip over the range of operating conditions.

Typically, all delays will be less than their maximum, with some delays being disproportionately faster than others. The 70% rule describes the spread in the scaling factors; the delay that decreases the most will be no less than 70% of what it would have been if it had scaled in proportion to the delay that decreased the least. In particular, in a worst-case design where it is assumed that any delay might not have scaled at all, and remains at its maximum value, other delays will be no less than 70% of their maximum.

Outputs

All XC3000/XC3100 LCA outputs are true CMOS with n-channel transistors pulling down and p-channel transistors pulling up. Unloaded, these outputs pull rail-to-rail. Some additional ac characteristics of the output are listed in Table 2. Figures 4 and 5 show output current/voltage curves for typical XC3000 and XC3100 devices. See other product family output characteristics on pages 8-6, 8-7 and 9-23.

Output-short-circuit-current values are given only to indicate the capability to charge and discharge capacitive loads. In accordance with common industry practice for other logic devices, only one output at a time may be short circuited, and the duration of this short circuit to V_{CC} or ground may not exceed one second. Xilinx does not recommend a continuous output or clamp current in

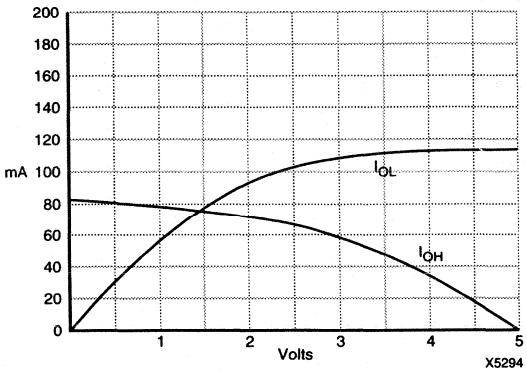


Figure 4. Output Current/Voltage Characteristics for XC3020

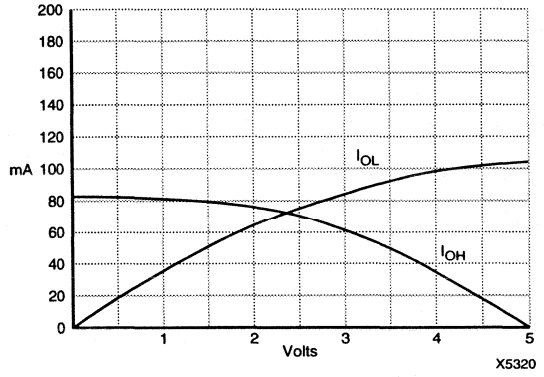


Figure 5. Output Current/Voltage Characteristics for XC3142

excess of 20 mA on any one output pin. The data sheet guarantees the outputs for no more than 4 mA at 320 mV to avoid problems when many outputs are sinking current simultaneously.

The active-High 3-state control (T) is the same as an active-Low output enable ($\overline{OE}$). In other words, a High on the T-pin of an OBUFZ places the output in a high impedance state, and a Low enables the output. The same naming convention is used for TBUFs within the LCA device.

I/O Clocks

Internally, up to eight distinct I/O clocks can be used, two on each of the four edges of the die. While the IOB does not provide programmable clock polarity, the two clock lines serving an IOB can be used for true and inverted clock, and the appropriate polarity connected to the IOB. This does, however, limit all IOBs on that edge of the die to using only the two edges of the one clock.

IOB latches have active-Low Latch Enables; they are transparent when the clock input is Low and are closed when it is High. The latch captures data on what would otherwise be the active clock edge, and is transparent in the half clock period before the active clock edge.

Table 2. Additional AC Output Characteristics

AC Parameters	Fast*	Slow*
Unloaded Output Slew Rate	2.8 V/ns	0.5 V/ns
Unloaded Transition Time	1.45 ns	7.9 ns
Additional rise time for 812 pF	100 ns	100 ns
normalized	0.12 ns/pF	0.12 ns/pF
Additional fall time for 812 pF	50 ns	64 ns
normalized	0.06 ns/pF	0.08 ns/pF

* Fast and Slow refer to the output programming option.

Routing

Horizontal Longlines

As shown in Table 3, there are two horizontal Longlines (HLLs) per row of CLBs. Each HLL is driven by one TBUF for each column of CLBs, plus an additional TBUF at the left end of the Longline. This additional TBUF is convenient for driving IOB data onto the Longline. In general, the routing resources to the T and I pins of TBUFs are somewhat limited.

Optionally, HLLs can be pulled up at either end, or at both ends. The value of each pull-up resistor is 3-10 kΩ.

In addition, HLLs are permanently driven by low-powered latches that are easily overridden by active outputs or pull-up resistors. These latches maintain the logic levels on HLLs that are not pulled up and temporarily are not driven. The logic level maintained is the last level actively driven onto the line.

When using 3-state HLLs for multiplexing, the use of fewer than four TBUFs can waste resources. Multiplexers with four or fewer inputs can be implemented more efficiently using CLBs.

Table 3. Number of Horizontal Longlines

Part Name	Rows x Columns	CLBs	Horizontal Longlines	TBUFs per HLL
XC3020	8 x 8	64	16	9
XC3030	10 x 10	100	20	11
XC3042	12 x 12	144	24	13
XC3064	16 x 14	224	32	15
XC3090	20 x 16	320	40	17
XC3195	22 x 22	484	44	23

Vertical Longlines

There are four vertical Longlines per routing channel: two general purpose, one for the global clock net and one for the alternate clock net.

Clock Buffers

XC3000/XC3100 devices each contain two high-fan-out, low-skew clock-distribution networks. The global-clock net originates from the GCLK buffer in the upper left corner of the die, while the alternate clock net originates from the ACLK buffer in the lower right corner of the die.

The global and alternate clock networks each have optional fast CMOS inputs, called TCLKIN and BCLKIN, respectively. Using these inputs provides the fastest path from the PC board to the internal flip-flops and latches. Since the signal bypasses the input buffer, well-defined CMOS levels must be guaranteed on these clock pins.

To specify the use of TCLKIN or BCLKIN in a schematic, connect an IPAD symbol directly to the GCLK or ACLK symbol. Placing an IBUF between the IPAD and the clock buffer will prevent TCLKIN or BCLKIN from being used.

The clock buffer output nets only drive CLB and IOB clock pins. *They do not drive any other CLB inputs.* In rare cases where a clock needs to be connected to a logic input or a device output, a signal should be tapped off the clock buffer input, and routed to the logic input. This is not possible with clocks using TCLKIN or BCLKIN.

The clock skew created by routing clocks through local interconnect makes safe designs very difficult to achieve, and this practice is not recommended. In general, the fewer clocks that are used, the safer the design. High fan-out clocks should always use GCLK or ACLK. If more than two clocks are required, the ACLK net can be segmented into individual vertical lines that can be driven by PIPs at the top and bottom of each column. Clock signals routed through local interconnect should only be considered for individual flip-flops.

General Information

Recovery from Reset

Recovery from Reset is not specified in Xilinx data sheets because it is very difficult to measure in a production environment. The following values may be assumed for all XC3000/XC3100 devices and speed grades.

- The CLB can be clocked immediately (<0.2 ns) after the end of the internal Reset Direct signal (RD).
- The CLB can be clocked no earlier than 25 ns (worst case) after the release of an externally applied Global Reset signal, i.e., after the rising edge of the active-Low signal.

Configuration and Start-up

Until the chip goes active after configuration, all I/O pins not involved in the configuration process remain in a high-impedance state with weak pull-up resistors; all internal flip-flops and latches are held reset. Multiple LCA devices hooked up in a daisy chain will all go active simultaneously on the same CCLK edge. This is well documented in the data sheets.

Not documented, however, is how the internal combinatorial logic comes alive during configuration: As configuration data is shifted in and reaches its destination, it activates the logic and also "looks at" the IOB inputs. Even the crystal oscillator starts operating as soon as it receives its configuration data. Since all flip-flops and latches are being held reset, and all outputs are being held in their high-impedance state, there is no danger in this "staggered awakening" of the internal logic. The operation of the logic prior to the end of configuration is even useful; it ensures that clock enables and output enables are correctly defined before the elements they control become active.

Once configuration is complete, the LCA device is activated. This occurs on a rising edge of CCLK, when all outputs and clocks that are enabled become active simultaneously. Since the activation is triggered by CCLK, it is an asynchronous event with respect to the system clock. To avoid start-up problems caused by this asynchronism, some designs might require a reset pulse that is synchronized to the system clock.

The circuit shown in Figure 5 generates a short Global Reset pulse in response to the first system clock after the end of configuration. It uses one CLB and one IOB, and also precludes the use of the $\overline{\text{LDC}}$ pin as I/O.

During Configuration, $\overline{\text{LDC}}$ is asserted Low and holds the D-input of the flip-flop High, while Q is held Low by the internal reset, and $\overline{\text{RESET}}$ is kept High by internal and external pull-up resistors. At the end of configuration, the $\overline{\text{LDC}}$ pin is unasserted, but D remains High since the function generator acts as an R-S latch; Q stays Low, and $\overline{\text{RESET}}$ is still pulled High by the external resistor. On the first system clock after configuration ends, Q is clocked High, resetting the latch and enabling the output driver. which forces $\overline{\text{RESET}}$ Low. This resets the whole chip until the Low on Q permits $\overline{\text{RESET}}$ to be pulled High again.

The whole chip has thus been reset by a short pulse instigated by the system clock. No further pulses are generated, since the High on $\overline{\text{LDC}}$ prevents the R-S latch from becoming set.

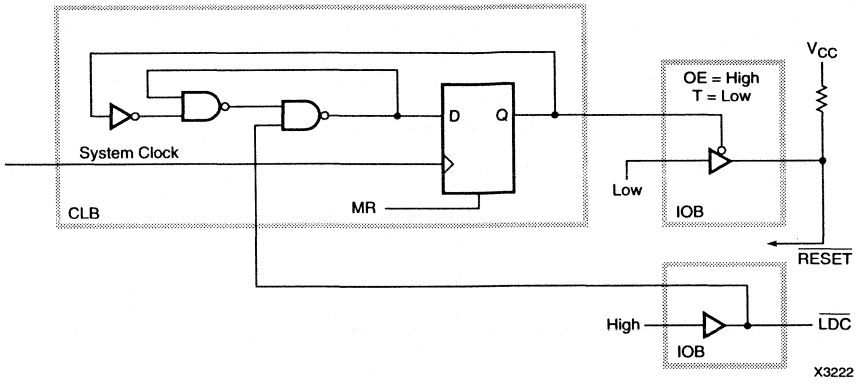


Figure 5. Synchronous Reset

Power Dissipation

As in most CMOS ICs, almost all LCA power dissipation is dynamic, and is caused by the charging and discharging of internal capacitances. Each node in the device dissipates power according to the capacitance in the node, which is fixed for each type of node, and the frequency at which the particular node is switching, which can be different from the clock frequency. The total dynamic power is the sum of the power dissipated in the individual nodes.

While the clock line frequency is easy to specify, it is usually more difficult to estimate the average frequency of other nodes. Two extreme cases are binary counters, where half the total power is dissipated in the first flip-flop, and shift registers with alternating zeros and ones, where the whole circuit is exercised at the clocking speed.

A popular assumption is that, on average, each node is exercised at 20% of the clock rate; a major EPLD vendor uses a 16-bit counter as a model, where the effective percentage is only 12%. Undoubtable, there are extreme cases, where the ratio is much lower or much higher, but 15 to 20% may be a valid approximation for most normal designs. Note that global clock lines must always be entered with their real, and obviously well-known, frequency.

Consequently, most power consumption estimates only serve as guidelines based on gross approximations. Table 4 shows the dynamic power dissipation, in mW per MHz, for different types of XC3000 nodes. While not precise, these numbers are sufficiently accurate for the calculations in which they are used, and may be used for any XC3000/XC3100 device. Table 5 shows a sample power calculation.

Table 4. Dynamic Power Dissipation

	XC3020	XC3090	
One CLB driving three local interconnects	0.25	0.25	mW/MHz
One device output with a 50 pF load	1.25	1.25	mW/MHz
One Global Clock Buffer and line	2.00	3.50	mW/MHz
One Longline without driver	0.10	0.15	mW/MHz

Table 5. Sample Power Calculation for XC3020

Quantity	Node		MHz	mW/MHz	mW
	Clock Buffer	1	40	2.00	80
	CLBs	5	40	0.25	50
	CLBs	10	20	0.25	50
	CLBs	40	10	0.25	100
	Longlines	8	20	0.10	16
	Outputs	20	20	1.25	500
				Total Power	~800

Crystal Oscillator

XC3000 and XC3100 devices contain an on-chip crystal oscillator circuit that connects to the ACLK buffer. This circuit, Figure 6, comprises a high-speed, high-gain inverting amplifier with its input connected to the dedicated XTL2 pin, and its output connected to the XTL1 pin. An external biasing resistor, R1, with a value of 0.5 to 1 MΩ is required.

A crystal, Y1, and additional phase-shifting components, R2, C1 and C2, complete the circuit. The capacitors, C1 and C2, in parallel form the load on the crystal. This load is specified by the crystal manufacturer, and is typically 40 pF. The capacitors should be approximately equal: 20 pF each for a 40 pF crystal.

Either series- or parallel-resonant crystals may be used, since they differ only in their specification. Crystals constrain oscillation to a narrow band of frequencies, the width of which is <<1% of the oscillating frequency; the exact frequency of oscillation within this band depends on the components surrounding the crystal. Series-resonant crystals are specified by their manufacturers according to the lower edge of the frequency band, parallel-resonant crystals according to the upper edge.

The resistor R2 controls the loop gain and its value must be established by experimentation. If it is too small, the oscillation will be distorted; if it is too large, the oscillation will fail to start, or only start slowly. In most cases, the value of R2 is non-critical, and typically is 0 to 1 kΩ.

Once the component values have been chosen, it is good practice to test the oscillator with a resistor (~1 kΩ) in series with the crystal. If the oscillator still starts reliably, independent of whether the power supply turns on quickly or slowly, it will always work without the resistor.

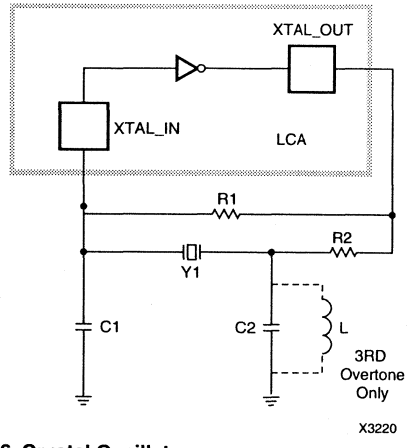


Figure 6. Crystal Oscillator

Table 6. Third Harmonic Crystal Oscillator Tank-Circuit

Frequency	LC Tank				
(MHz)	L (μH)	C2 (pF)	Freq (MHz)	R2 (Ω)	C1 (pF)
32	1	60	20.6	430	23
35	1	44	24.0	310	23
49	1	31	28.6	190	23
72	1	18	37.5	150	12

For operation above 20 to 25 MHz, the crystal must be operated at its third harmonic. The capacitor C2 is replaced by a parallel-resonant LC tank circuit tuned to ~2/3 of the desired frequency, i.e., twice the fundamental frequency of the crystal. Table 6 shows typical component values for the tank circuit.

See pages 9-30 and 9-31 for a more detailed description of crystal oscillators.

CCLK Frequency Variation

The on-chip R-C oscillator that is brought out as CCLK also performs several other internal function. It generates the power-on delay, 2^{16} = 65,536 periods for a master, 2^{14} = 16,384 periods for a slave or peripheral device. It generates the shift pulses for clearing the configuration array, using one clock period per frame, and it is the clock source for several small shift registers acting as low-pass filters for a variety of input signals.

The nominal frequency of this oscillator is 1 MHz with a max deviation of +25% to -10%. The clock frequency, therefore, is between 1.25 MHz and 0.5 MHz. In the XC4000 family, the 1-MHz clock is derived from an internal 8-MHz clock that also can be used as CCLK source.

Xilinx circuit designers make sure that the internal clock frequency does not get faster as devices are migrated to smaller geometries and faster processes. Even the newest and fastest Xilinx FPGA is compatible with the oldest and slowest device ever manufactured. The CCLK frequency is fairly insensitive to changes in V_{CC}, varying only 0.6% for a 10% change in V_{CC}. It is, however, very temperature dependent, increasing 40% as the temperature drops from 25°C to -30°C, Table 7.

Table 7. CCLK Frequency Variation on a Sample Device

V_{CC}	Temp	Frequency
4.5 V	25°C	687 kHz
5.0 V	25°C	691 kHz
5.5 V	25°C	695 kHz
4.5 V	-30°C	966 kHz
4.5 V	+130°C	457 kHz

Metastable Recovery

Whenever a clocked flip-flop synchronizes an asynchronous input, there is a small probability that the flip-flop output will exhibit an unpredictable delay. This happens when the input transition not only violates the setup and hold-time specifications, but actually occurs within the tiny timing window where the flip-flop accepts the new input. Under these circumstances, the flip-flop can enter a symmetrically balanced transitory state, called metastable (meta = between).

While the slightest deviation from perfect balance will cause the output to revert to one of its two stable states, the delay in doing so depends not only on the gain bandwidth product of the circuit, but also on how perfect the balance is, and on the noise level within the circuit; the delay can, therefore, only be described in statistical terms.

The problem for the system designer is not the illegal logic level in the balanced state (it's easy enough to translate that to either a 0 or a 1), but the unpredictable timing of the final change to a valid logic state. If the metastable flip-flop drives two destinations with differing path delays, one destination might reflect the final data state while the other does not.

With the help of a mostly self-contained circuit on the demonstration board that is available to all Xilinx customers, Xilinx evaluated the XC3020-70 CLB flip-flop. The result of this evaluation shows the Xilinx CLB flip-flop to be superior in metastable performance to many popular MSI and PLD devices.

Statistically, when an asynchronous event with a frequency of approximately 1 MHz is being synchronized by a 10-MHz clock, the CLB flip-flop suffers an additional delay, of 4.2 ns once per hour, and 8.4 ns once per 1,000 years.

The frequency of occurrence of these metastable delays is proportional to the product of the asynchronous event frequency and the clock frequency. If, as an example, a 100-kHz event is synchronized by a 2-MHz clock, the above delays (besides being far more tolerable) will occur 50 times less often.

Since metastability can only be measured statistically, this data was obtained by configuring an XC3020 with eight concurrent detectors. Eight D-type flip-flops were clocked from a common high-speed source, and their D inputs driven from a common, lower frequency asynchronous signal, Figure 7. The output of each flip-flop fed the D inputs of two more flip-flops, one clocked half a clock period later and the second a full clock period later.

If a metastable event in the first flip-flop increased the output settling time to more than one-half clock period, the second two flip-flops would capture differing data. Thus, the occurrence of a long metastable delay could be detected using a simple comparator. Deliberate skew in the input data to the eight metastable circuits ensured that at most one metastable event could occur each clock. This permitted the eight detectors to be ORed into a single metastable event counter.

As expected, no metastable events were observed at clock rates below 25 MHz, since a half clock period of 20 ns is adequate for almost any metastability-resolution delay plus the flip-flop set-up time. Increasing the clock rate to around 27 MHz brought a sudden burst of metastable events. Careful adjustment of the clock frequency gave repeatable, reliable measurements showing that a 500 ps decrease in the half clock period increased the frequency of metastable occurrences by a factor of 41.

To be conservative, to compensate for favorable conditions at room temperature and to avoid any possibility of overstating a good case, the measurements were interpreted as follows:

When capturing asynchronous data, the error rate decreases by a factor of 40 for every additional nanosecond of metastability-resolution delay that the system can tolerate.

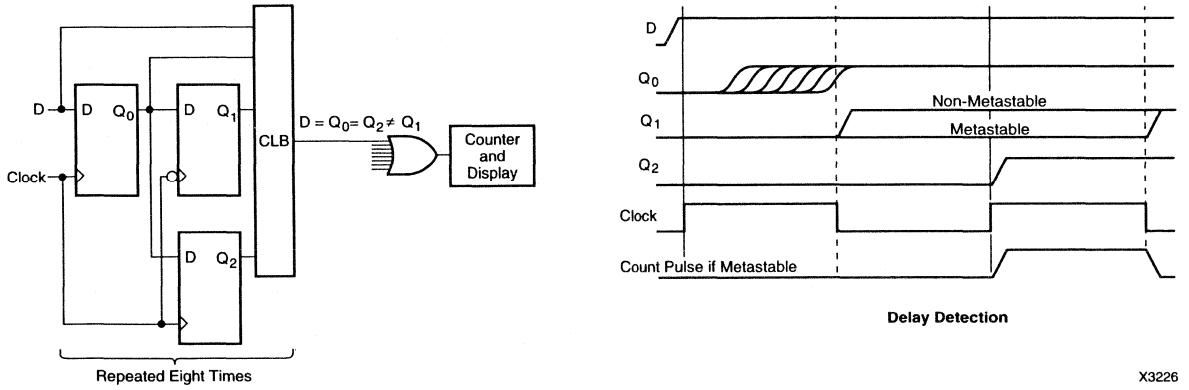


Figure 7. Metastable Measuring Circuit

Metastability Calculations

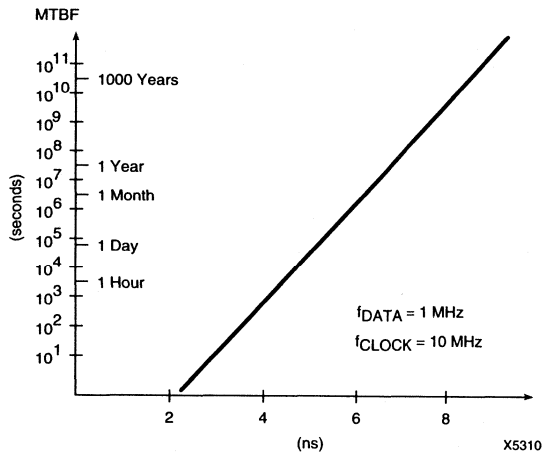
The Mean Time Between Failure (MTBF) can only be defined statistically. It is inversely proportional to the product of the two frequencies involved, the clock frequency and the average frequency of data changes, provided that these two frequencies are independent and have no correlation.

K1 is a factor that has the dimension of time, and describes the likelihood of going metastable. K2 is an exponent that describes the speed with which the metastable condition is being resolved.

$1/MTBF = f1 \bullet f2 \bullet K1 \bullet e^{-K2 \times t}$

MTBF in seconds
f1 and f2 in Hz
K1 = 1.5 • 10-10 seconds (measured for XC3020-70)
K2 = ln 40 / ns = 3.69 • 10^9 per second (XC3020-70)

For a 10 MHz clock and approximately 1 MHz data rate, the table below gives the expected MTBF as a function of the acceptable extra delay at the output of the metastable-going flip-flop.



Extra Delay	MTBF
1.0 ns	27 milliseconds
4.2 ns	1 hour
6.7 ns	423 days
8.5 ns	890 years
10.0 ns	225,000 years
11.0 ns	9 million years
12.0 ns	360 million years

Figure 8. Metastable MTBF as a Function of Additional Acceptable Delay

Battery Back-up

Since Logic Cell Arrays are manufactured using a high-performance low-power CMOS process, they can preserve the configuration data stored in the internal static memory cells even during a loss of primary power. This is accomplished by forcing the device into a low-power non-operational state, while supplying the minimal current requirement of V_{CC} from a battery.

Circuit techniques used in XC3100 and XC4000 devices prevent I_{CC} from being reduced to the level need for battery back-up. Consequently, battery back-up should only be used for XC2000, XC3000 and XC3000A devices.

There are two primary considerations for battery backup which must be accomplished by external circuits.

- Control of the Power-Down ($\overline{PWRDWN}$) pin
- Switching between the primary V_{CC} supply and the battery.

Important considerations include the following.

- Insure that $\overline{PWRDWN}$ is asserted logic Low prior to V_{CC} falling, is held Low while the primary V_{CC} is absent, and returned High after V_{CC} has returned to a normal level. $\overline{PWRDWN}$ edges must not rise or fall slowly.
- Insure "glitch-free" switching of the power connections to the LCA device from the primary V_{CC} to the battery and back.
- Insure that, during normal operation, the LCA V_{CC} is maintained at an acceptable level, 5.0 V ± 5% (±10% for Industrial and Military).

Figure 9 shows a power-down circuit developed by Shel Epstein of Epstein Associates, Wilmette, IL. Two Schottky diodes power the LCA from either the 5.2 V primary supply or a 3 V Lithium battery. A Seiko S8054 3-terminal power monitor circuit monitors V_{CC} and pulls $\overline{PWRDWN}$ Low whenever V_{CC} falls below 4 V.

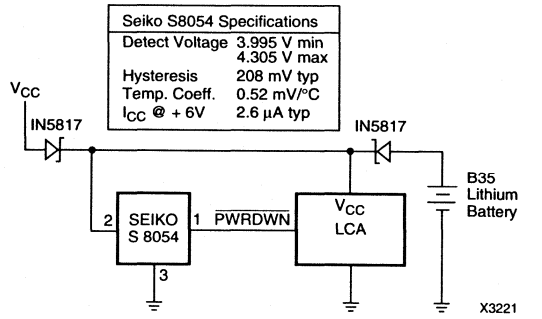


Figure 9. Battery Back-up Circuit

Improving XC4000 Design Performance

XAPP 043.000

Application Note BY NICK CAMILLERI AND CHRIS LOCKHARD

Summary

This Application Note describes XC4000 architectural features that can be exploited in high-performance designs, and software techniques that improve placement, routing and timing. It also contains information necessary for advanced design techniques, such as floor planning, locking down I/Os, and critical path optimization.

LCA Family	*Demonstrates*
XC4000	High-performance XC4000 design

Introduction

Designers sometimes assume that the Xilinx FPGA architecture is a gate-array-like sea-of-gates and that, consequently, little or no architectural consideration is required during design. This approach is valid and is supported by XMAKE, the Xilinx fully-automated design procedure. It can, however, lead to inefficient designs.

The Xilinx FPGA architecture is very regular and gate-array-like, but it is not a simple sea-of-gates. An understanding of the architecture and the resources it provides can make designs become more efficient in both speed and density. This Application Note focuses on the features of the XC4000 architecture and its supporting software that improve design efficiency. It also describes advanced design techniques that extract the maximum performance from the architecture.

Some techniques described in this Application Note relate specifically to XACT v1.42. In a subsequent version, XACT 5, Hard Macros will be replaced by Relationally Placed Macros, and the operation of XACT Performance will change significantly.

XC4000 Architectural Features

XC4000 CLB Overview

The XC4000 CLB is shown in Figure 1. Key features are the three function generators and the two flip-flops. Unlike previous LCA devices, the F and G function generators do not share inputs, permitting them to operate totally independently, if required. The H function generator combines the F and G outputs with an additional H1 input.

The F-G-H combination can implement any function of five inputs. In addition, some functions of more inputs can also be implemented. Some functions of five inputs can be implemented using just an F-H or G-H combination.

The two flip-flops can store the function-generator outputs or a signal coming in on the DIN pin. If the H function generator is not in use, the H1 input can pass through the function generator and provide a second direct input to the other flip-flop. Since separate pairs of output pins are provided for the function generators and the flip-flops, the F and G function generators and the two flip-flops can operate independently.

Fast Carry Logic

In addition to implementing logic and providing storage, the XC4000 CLB contains dedicated hardware to accelerate the carry path of adders and counters, Figure 2. Using this feature, adders and counters are very fast and efficient, consuming a minimum number of CLBs.

While dedicated logic and interconnect are used to optimize the carry path, function generators are used to form sums from the operands and carries. In this way, two bits of arithmetic (or one bit, if so desired) can be implemented in each CLB. Since the dedicated carry logic can be configured in approximately 40 different ways, CLBs can be concatenated into a variety of arithmetic functions. The carry propagates either up or down a column of CLBs.

The dedicated carry logic is accessible via hard macros. Carry-logic hard macros are available in the Xilinx library, or may be defined by the user with a program called HMGEN*. The HMGEN package includes documentation which describes how to use both the HMGEN program and the dedicated carry logic.

For additional information on the dedicated carry logic see the Xilinx Application Notes Using the Dedicated Carry Logic in the XC4000 (XAPP 013) and Estimating the Performance of XC4000 Adders and Counters (XAPP 018).

*HMGEN is available free of charge from Xilinx. Contact the Xilinx Technical Support Hotline at 1-800-255-7778

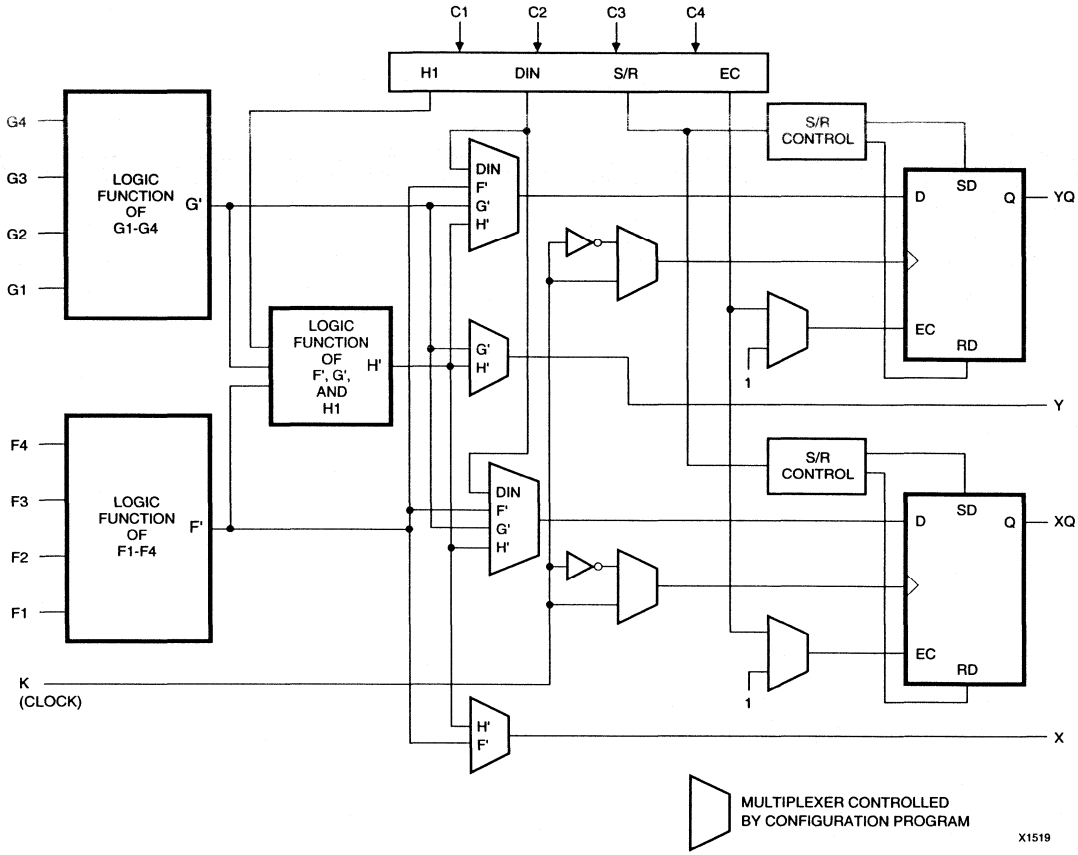


Figure 1. XC4000 Configurable Logic Block

On-Chip RAM

The XC4000 on-chip RAM significantly reduces the cost of data storage. Using this feature, up-to-64 bits of data can stored in a single CLB that otherwise could only store two bits in its flip-flops.

Any CLB can be configured as a RAM. In the RAM mode, the F- and G-function-generator look-up tables become writable, Figure 3. For a 16 x 2-bit RAM, the F and G function generators are used separately. For a 32 x 1-bit RAM, they are combined in the H function generator.

The F- and G-input pins act as memory address lines, just as they do in the non-RAM mode. Other CLB control pins, however, are redefined. For a 16 x 2-bit RAM, DIN and H1 become the two data inputs, while in the 32 x 1-bit configuration, DIN is the single data input, and D1 the fifth address bit. In both cases, S/R is the Write Enable (WE) input.

In the 16 x 2-bit mode, read data is available at the F- and G-function-generator outputs; in the 32 x 1-bit mode, it is available at the H-function-generator output. Just as in the non-RAM mode, these function-generator outputs drive the X and Y output pins, or they can be registered in the flip-flops.

Some non-RAM functionality remains when CLBs are configured as RAM. In the 16 x 2-bit mode, the H function generator can be used to implement any function of the two RAM outputs. In the 32 x 1-bit mode, both the write and the read data can be captured in the flip-flops, since the flip-flops have access to the RAM input data on the DIN pin.

The XC4000 RAM function is extremely fast compared to monolithic SRAM devices that often have cycle times of 55 ns or longer. In those slower devices, 1 ns glitches in control signals can be tolerated. This is not the case in the XC4000 RAM, however, where cycle times are less than 10 ns. WE pulses as short as 1 ns are easily recognized, and good control-circuit design is essential.

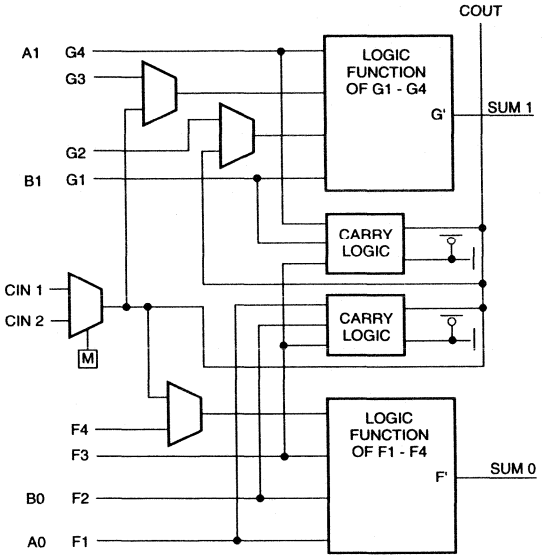


Figure 2. XC4000 Fast Carry Logic in Each CLB

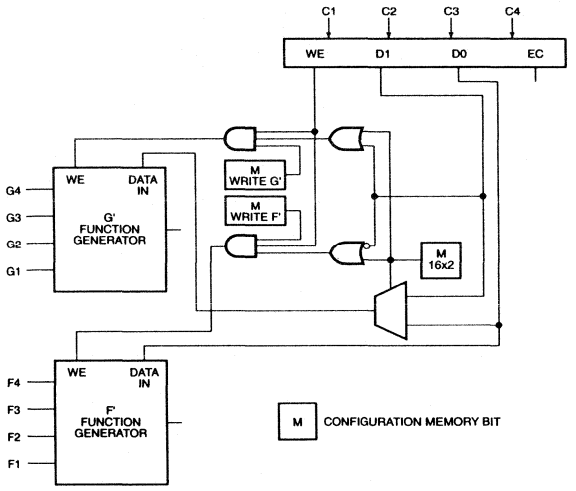


Figure 3. XC4000 CLB RAM Mode

Designing with the XC4000 SRAM is similar to designing with very fast monolithic SRAMs (<25 ns cycle time). Many factors, such as interconnect delays, that can safely be ignored in slower monolithic SRAM designs become critical. For a discussion of XC4000 RAM design, please refer to the Xilinx Application Notes Using the XC4000 RAM Capability (XAPP 031) and High-Speed RAM design in XC4000 (XAPP 042).

3-State Buffers

To facilitate on-chip multiplexed busses, the XC4000 architecture includes Longlines that can be driven by three-state buffers (TBUFs), Figure 4. Two TBUFs, located adjacent to each CLB, drive the horizontal Longlines immediately above and below the CLB.

The TBUF data inputs can easily be driven from the outputs of the associated CLB, but can also come from elsewhere. While any signal can be used to enable the TBUFs, if an enable is routed on a vertical Longline, it can be used to select a column-wise function to drive a horizontal bus.

Additional TBUFs are located near the Input/Output blocks (IOBs) on the left- and right-hand edges of the array. These permit IOBs to be included in a multiplexed bus, thus possibly extending an external bi-directional bus onto the chip.

TBUF Longlines can also be used to implement wired-AND functions. Optional resistive pull-ups at each end of a Longline cause it to go High while not being driven. Enabling any TBUF that has a logic Low at its input will cause the line to go Low, thus creating a wired-AND of the enable signals.

The TBUF Longlines are valuable routing resources, and should be used sparingly. Multiplexers and AND-gates with 16 or fewer inputs can often be implemented more efficiently using CLBs, thereby conserving Longline resources. TBUFs are rarely appropriate for multiplexers with four or fewer inputs.

Global Clock Buffers

Eight global nets run through XC4000 devices, potentially reaching every CLB, Figure 5. These global nets are optimized for the distribution of clocks and other time-critical or high-fanout signals.

Four of the eight are primary global nets that offer minimum delay and negligible skew. The other four are secondary global nets. Due to heavier loading, the secondary global nets introduce a slightly longer delay, about 1 ns more, and some additional skew.

Primary and secondary global nets are driven by BUFGP and BUFGS buffers, respectively, both of which can connect directly to pads. The use of a global net is specified in the schematic by using the appropriate buffer to drive the desired signal. BUFGP and BUFGS symbols are in the Xilinx library.

Switch matrices at the center of each column connect the eight global buffers to four vertical lines used to distribute global signals within the column. Figure 6 shows one of the switch matrices. Each vertical line can be driven by only one BUFGP, and each BUFGP drives a different vertical line. The BUFGSs, on the other hand, can each drive any or all vertical lines.

Consequently, BUFGSs are much more flexible in their routing. This flexibility is particularly valuable when routing

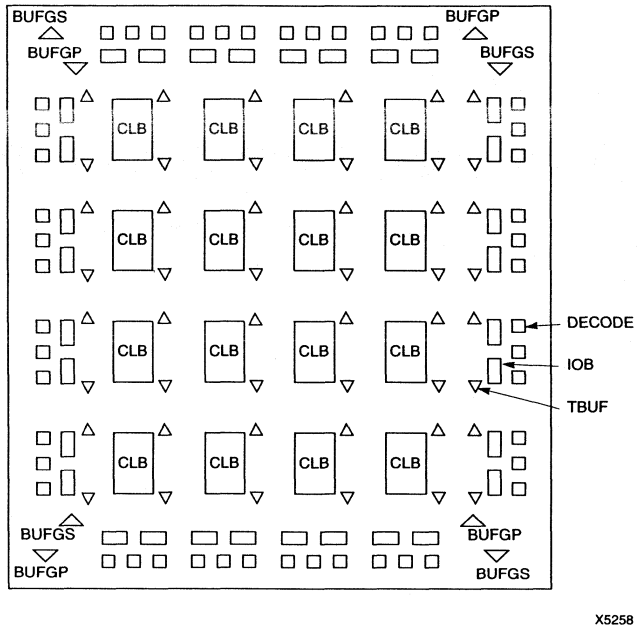


Figure 4. XC4000 TBUF Organization

non-clock global signals, since the vertical lines have limited connections to non-clock CLB pins. Being able to drive any, or even multiple vertical lines from a single buffer is a tremendous advantage.

The routing of non-clock global signals also benefits by using BUFGSs for clock distribution. A BUFGP would constrain the clock routing to the same vertical line in every column. With a BUFGS, however, any vertical line can be used for the clock, possibly freeing a critical line for non-clock routing. ***Whenever timing and skew requirements permit, BUFGSs should be used for distributing global signals.***

The number of clocks in a system should be minimized. Since only four clocks can be made available in a column of CLBs, a large number of clocks imposes considerable constraints on CLB placement. In particular, gated clocks should be avoided, unless absolutely necessary. In addition to consuming global nets, the gating logic introduces uncontrolled clock skew and the potential for clock glitches, both of which can cause a system to malfunction.

It is better to use a minimal number of clocks and disable the flip-flops when clocking is not required. Clock-enable signals can often be routed on local interconnect or regular Longlines, since they are not skew-critical.

Wide Decoders

Sometimes it is necessary to decode specific values from a large number of bits, e.g., when decoding a specific microprocessor memory address. To facilitate such decoding, dedicated wide-decoder functions are provided along each edge of XC4000 devices, Figure 7. These wide decoders are separate from the CLBs, and do not consume CLB resources.

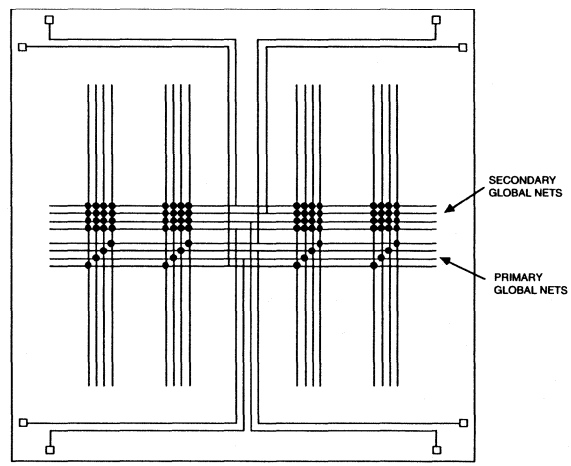


Figure 5. XC4000 Global Net Distribution

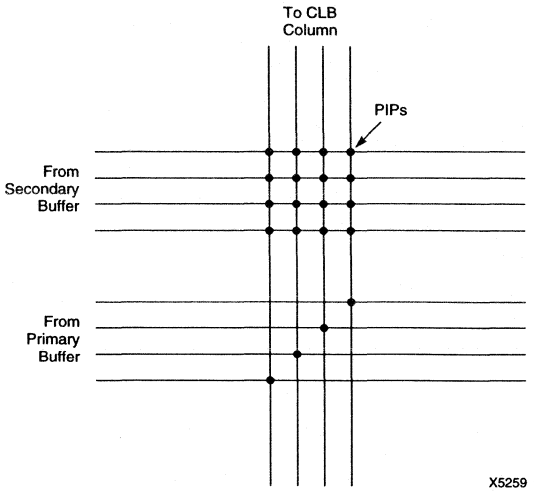


Figure 6. XC4000 Global Net Interconnection Matrix

On each edge of the chip, there are four decoders that share a common set of inputs. Potentially, there are three decoder inputs for each row or column of CLBs, one each from two adjacent IOBs and one from local interconnect. While the decoders share inputs, it is possible to decode different sets of bits on the same edge, since all inputs need not be used in every decoder. It is thus even possible to decode disjoint sets of bits on the same edge.

An XC4000, for example, has a 20 x 20 array of CLBs, and each edge has four wide decoders sharing 60 inputs. The decoders on one edge could decode a specific byte address and a specific word address from a 32-bit microprocessor bus, and, at the same time, decode two specific values from an internal 16-bit bus.

The wide decoders are implemented as wired-AND-gates. Resistive pull-ups at each end cause the output to go High when all inputs meet their specified condition. Any input to the decoder that fails to meet its specified condition (High or Low) causes the output to be pulled Low.

Each decoder can be split at its center to make two half-sized decoders. There are, therefore, up to 32 decoders available. Dedicated decoders should only be used to decode ten or more inputs, since nine or fewer inputs can be decoded more efficiently using a single CLB.

Note that XC4000A devices have two wide decoders per edge, instead of the four found in non-A devices. The use of wide decoders is specified in the schematic by using the symbols DECODE4, DECODE 8, etc.

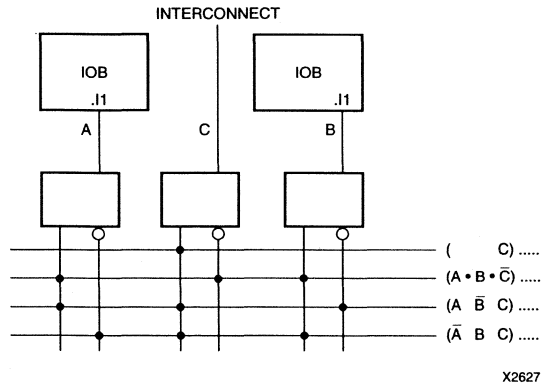


Figure 7. XC4000 Wide Decoder

IOB Registers

The latches and registers located in the IOBs are an often-overlooked resource, Figure 8. They are ideal for synchronizing input and output signals, and can also provide additional storage for internal signals. Using IOB registers can significantly reduce CLB flip-flop utilization, and can also reduce routing congestion.

The use of IOB registers or latches is specified during design entry. They are represented by the schematic symbols OUTFF, INFF and INLAT.

Master Set/Reset.

XC4000 devices contain a global-set/reset (GSR) line. When GSR is asserted, every flip-flop in the LCA device is simultaneously set or reset. No general-purpose routing resources are consumed, however, since the GSR has its own dedicated routing. Setting or resetting all the flip-flops in this way is far more efficient that using the RD pins on individual CLBs.

Each flip-flop is either set or reset according to an attribute attached to it in the schematic. The default value for the

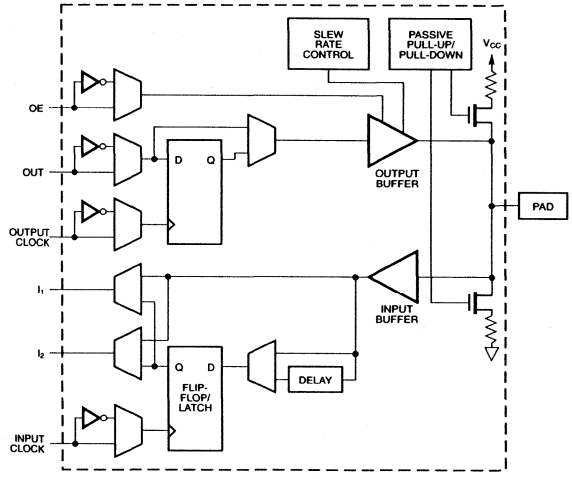


Figure 8. XC4000 Input/Output Block

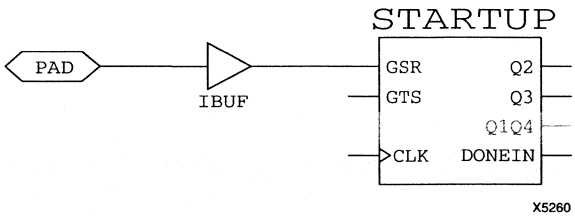


Figure 9. Schematic Symbols for Global Set/Reset

attribute is INIT=R, which causes GSR to reset the flip-flop. Changing the attribute to INIT=S causes the flip-flop to be set. The INIT attribute also determines the flip-flop state immediately after power-up.

To use GSR, the STARTUP primitive should be included in the schematic, and a pad connected to its GSR pin, Figure 9. Any user-I/O pin may be used, and, if necessary, the pad can be locked to a specific device pin, just like any other pad. If possible, however, PPR should be allowed to choose the location, since any unnecessary constraint reduces the freedom PPR has to implement the design, and potentially degrades the result.

Boundary-Scan Circuitry

In production, boards must be tested to assure the integrity of both the components and the interconnections. However, as integrated circuits become more complex and multi-layer PC boards become more dense, it is increasingly difficult to test assembled boards.

XC4000 helps solve this problem by providing boundary-scan test facilities. All user-I/O pins are fully testable, and the test protocols are compatible with IEEE Std 1149.1. Boundary scan does not detract from the capacity or capability of XC4000 devices, since it only uses dedicated logic.

External testing (EXTEST) is fully supported, and there is limited support for internal self-test. For more information on boundary scan in XC4000 devices, see the Xilinx Application Note Boundary Scan in XC4000 Devices (XAPP 017).

Advanced Design Guidelines

Design Partitioning

The first phase of the design implementation process is partitioning. LCA devices emulate logic using look-up tables, and it is necessary to divide the schematic into groups of gates that will fit into individual look-up tables. Inefficient partitioning can both decrease the performance of a design and cause it to use more CLBs than necessary.

Inefficient partitioning can have two causes. First, the logic may have been drawn in such a way that convenient partitioning boundaries do not exist. PPR does not split large gates into parts that can be absorbed into unused

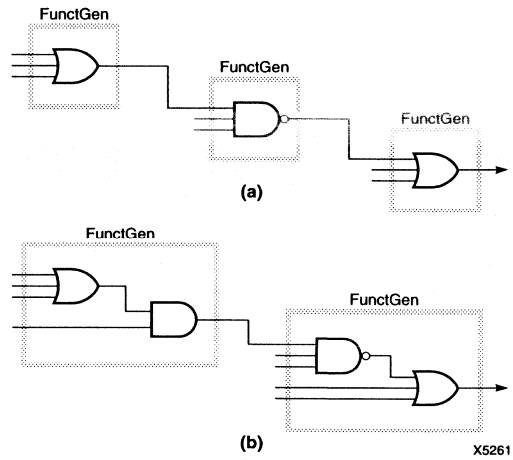


Figure 10. Improving Partitioning by Gate Decomposition

portions of surrounding function generators. Instead, it preserves gate boundary and may waste function generators. Figure 10 shows the same logic drawn in two ways; one way requires a cascade of three function generators, while the other way requires only two.

If a gate has a fanout greater than one, the partitioner always assigns the output of this gate to be the output of a function generator. PPR will not replicate the gate, even if the copies can be absorbed into other function generators. Figure 11 shows a second example where drawing the logic differently results in fewer CLBs and fewer levels of CLBs.

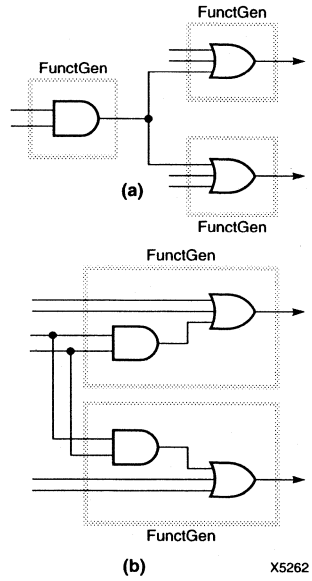


Figure 11. Improving Logic Partitioning by Gate Duplication

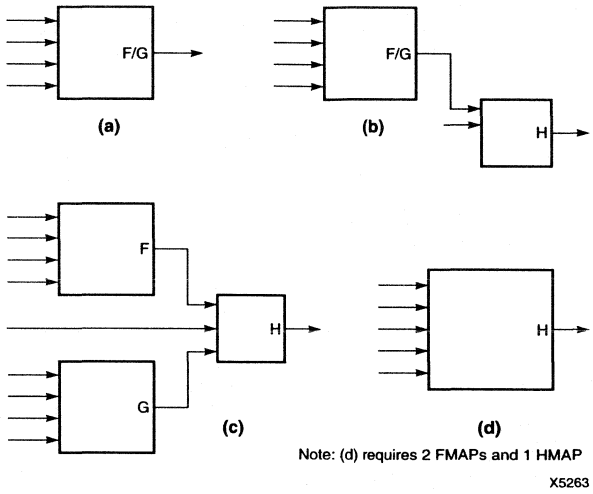


Figure 12. Preferred Functional-Block Sizes

Redrawing critical portions of an existing design can often improve its performance. It is much better, however, to draw the schematic initially in a way that guarantees a convenient partition. This is not as difficult as it might seem, and does not require gate-by-gate analysis or intimate knowledge of the partitioning algorithms.

Typically, the design process starts with a high-level block diagram, and progresses hierarchically through a series of increasingly detailed block diagrams as blocks are decomposed into smaller blocks. Eventually, the lowest-level blocks are translated into gates to create the logic diagram. The key to convenient partitioning is to structure the lowest level of blocks in such a way that it matches the XC4000 architecture.

Figure 12 shows four block structures that can conveniently be implemented in XC4000 devices. The four-input blocks correspond to F or G function generators, and the three-input blocks correspond to H function generators. The five-input block is a special case of the F-G-H combination. These blocks are characterized only by their number of inputs. Since a look-up table can emulate any function of its inputs, it is guaranteed that the logic required by each block will fit into a function generator, thus ensuring that a convenient partitioning exists.

This technique also facilitates performance estimates early in the implementation process. Each block in the block diagram corresponds to a delay specified in the XC4000 data sheet. With a simple routing allowance (add 50% to 100% of TILO per route), it is possible to estimate design feasibility. If necessary, structural design changes can be made before entering the schematic. For more advice on performance estimation, see to the Xilinx Application Note LCA Speed Estimation: Asking the Right Question (XAPP 011).

The second cause of apparent inefficient partitioning is the trade-off that PPR makes between area and speed. It is not possible for the built-in rule to match the requirements of every design. Partitioning can, however, be easily specified in the schematic using FMAPs and HMAPs, Figure 13.

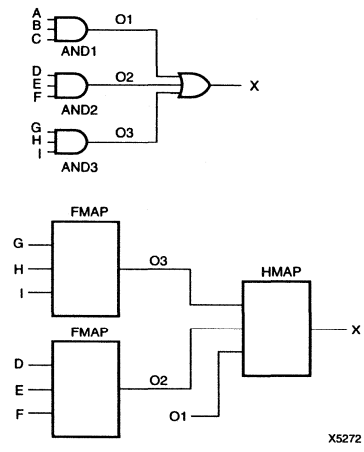


Figure 13. Using FMAPs and HMAPs

If an existing design is being optimized, isolated FMAPs and HMAPs can be added to improve critical paths. However, if the design technique described above is used, FMAPs and HMAPs can be included for the entire design to guarantee the expected partitioning. Little additional work is required since the FMAPs and HMAPs exactly match the blocks in the lowest-level block diagram.

Pipelining

Even with optimal partitioning, every logic function has a minimum achievable delay. If this minimum delay is too long, the function must be broken into smaller functions using pipelining.

Suppose a design needs to run at 20 MHz, and its worst-case Clock-to-Set-up delay is as shown in Table 1. Clearly, it does not meet its objective, and will only run up to 10 MHz. Of the 99.4 ns delay, 54.5 ns is attributable to block delays, and cannot be reduced by placement or routing improvements. In the unlikely event that all the routing delays could be reduced to the absolute minimum (~1.3 ns in XC4000-5), the worst-case Clock-to-Set-up delay of 66.2 ns would not still permit operation above 15 MHz. Using a more realistic routing estimate (50% to 100% of T_{ILO} per route), the design might run at only 10 – 13 MHz.

To achieve 20 MHz, there are four choices: use a faster device, improve the partitioning, restructure the logic to make the critical path less deep, or add a pipeline stage to break the function in two. While none to these may be possible in a particular case, pipelining is often the easiest choice.

Table 1. Worst-Case Clock-to-Set-Up Delay

Logical Path		Delay Cumulative
Source clock net : "CLK10MHZ" (Rising edge)		
From:BlkBLOCK1 CLOCK to CLB_R13C12.YQ:3.0 ns (3.0 ns)		
Thru: Net NET1	to CLB_R13C11.C4	: 3.8 ns (6.8 ns)
Thru: Blk BLOCK2	to CLB_R13C11.XQ	: 3.5 ns (10.3 ns)
Thru: Net NET2	to CLB_R23C11.F3	: 3.2 ns (13.5 ns)
Thru: Blk BLOCK3	to CLB_R23C11.X	: 4.5 ns (18.0 ns)
Thru: Net NET31	to CLB_R18C9.C2	: 13.3 ns (31.3 ns)
Thru: Blk BLOCK4	to CLB_R18C9.XQ	: 8.0 ns (39.3 ns)
Thru: Net NET4	to CLB_R13C10.F1	: 3.2 ns (42.5 ns)
Thru: Blk BLOCK5	to CLB_R13C10.Y	: 7.0 ns (49.5 ns)
Thru: Net NET5	to CLB_R2C11.C4	: 8.7 ns (58.2 ns)
Thru: Blk BLOCK6	to CLB_R2C11.YQ	: 8.0 ns (66.2 ns)
Thru: Net NET6	to CLB_R2C9.F3	: 3.7 ns (69.9 ns)
Thru: Blk BLOCK7	to CLB_R2C9.X	: 4.5 ns (74.4 ns)
Thru: Net NET7	to CLB_R3C10.F4	: 1.6 ns (75.9 ns)
Thru: Blk BLOCK8	to CLB_R3C10.X	: 4.5 ns (80.4 ns)
Thru: Net NET8	to CLB_R10C10.F3	: 4.8 ns (85.3 ns)
Thru: Blk BLOCK9	to CLB_R10C10.Y	: 7.0 ns (92.3 ns)
Thru: Net NET9	to CLB_R9C11.G1	: 2.6 ns (94.9 ns)
To: FF Setup (D), Blk BLOCK10		: 4.5 ns (99.4 ns)
Target FFX drives output net "NET10"		
Dest clock net : "CLK10MHZ" (Rising edge)		

In an LCA device, every function generator has a flip-flop at its output. If this flip-flop is not being used for other purposes, it can be inserted into the logic path with minimal layout changes. There are difficulties, however. The latency introduced by pipeline flip-flops must be matched elsewhere both in data and control signals.

In the example, a pipeline flip-flop at the output of BLOCK5 would be most effective. The worst-case Clock-to-Set-up for the two halves of the function is 52.9 ns. This still does not meet the 20 MHz target, but is close enough to expect that routing changes could complete the job.

Of course, it is much better to anticipate such problems than solve them when the design is almost complete. The design technique discussed in the previous section provides delay estimates that are more than adequate for the detection of gross problems. Pipelining can then be built into the design, and any timing complications handled much more easily.

Floorplanning

Structured designs with a datapath-like organization and fixed data width can benefit greatly from simple floorplanning. Designs that use multiplexed busses are also candidates. In this section, floorplanning is discussed with respect to the absolute placement of logic functions within an LCA device. Relative placement control is discussed afterwards, in the next section on Hard Macros.

FPGA floorplanning is similar to planning a schematic diagram; an organization that permits gates to be connected conveniently on the schematic usually permits them to be connected conveniently in the FPGA. It must be remembered, however, that busses have to run though logic, not around it.

In most cases, the design itself suggests the floor plan. To minimize routing, bits with the same weight should be aligned in rows, and functions should be organized for the best data flow. Alignment of bits in rows is particularly important if a multiplexed bus is required, since horizontal TBUF Longlines must used for the bus.

If no obvious structure can be exploited, Floorplanning should be avoided. Arbitrary or poorly chosen placement constraints can hinder PPR, and lead to a worse result than if PPR were to run unconstrained.

In many instances of floorplanning, it is only necessary to consider the relative position of bits and functions. When planning busses, however, the absolute location can also be significant.

As described earlier, there are two TBUF Longlines per column of CLBs. Consequently, the widest bus that can be accommodated has twice as many bits as there are CLB rows in the array. This assumes that the busses require full-length Longlines. If a bus can be implemented using half-length Longlines, the center splitter can be opened, to accommodate twice as many bits.

Even if there are enough CLB rows to use full-length Longlines, it is good practice to use half-length lines wherever possible. The shorter lines have less capacitance and are, therefore, faster. In addition, the unused Longline halves are a valuable resource that can be used for other purposes.

Vertical Longlines run across the TBUF Longlines, and are often used for control signals. These vertical lines are also splittable, and, again, it is best to use half-length lines wherever possible. Floor plans should try to confine data paths to a single quadrant of the array.

While it is possible to use both halves of a TBUF Longline for two bits of a bus, this situation is far from ideal. Control signals must be duplicated in two columns since the Longline halves do not overlap. Functions like arithmetic carry that run across the bus are also complicated if they split between the two halves of the array. It is much better to use a device that is large enough to contain the full data path in one half of the array.

There is an IOB at each end of every TBUF Longline, and these IOBs have good connectivity to the Longlines. Consequently, I/O pins that need to connect to internal busses should be placed on the left or right sides of the chip, rather than on the top or bottom edges, where they lead to unnecessary routing congestion.

If signals are constrained to specific pins, the bit order of the pins should match the bit order of the bus. Again, routing congestion will result if the bit orders differ. Xilinx Hard

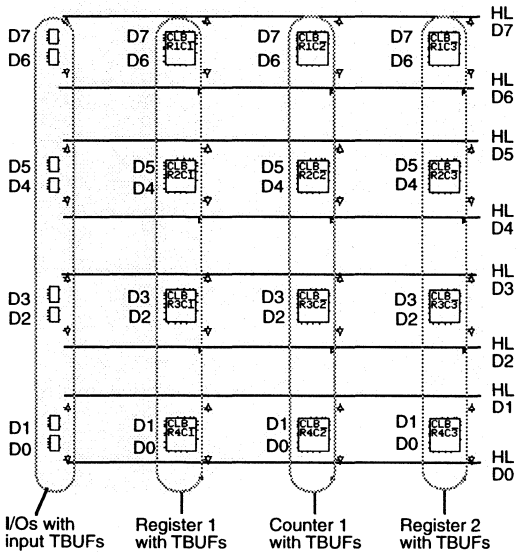


Figure 14. Bit Alignment for 3-State Bussing

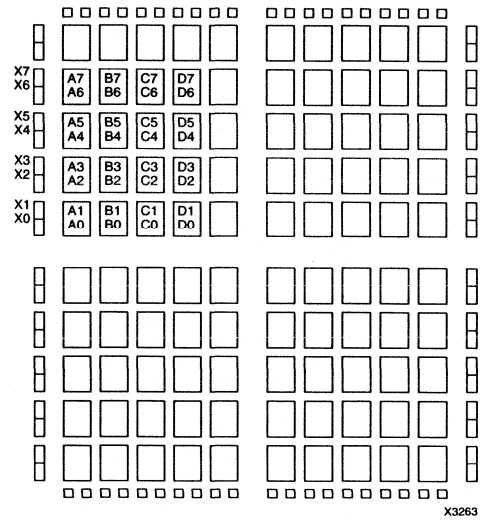


Figure 15. Efficient Placement in an XC4003

Macros have their bits in order with the MSB at the top, and if library macros are to be used, this standard should be adopted.

Consider the example shown in Figure 14. Two 8-bit registers, A[7:0] and B[7:0], and two 8-bit counters, C[7:0] and D[7:0], are multiplexed onto an 8-bit bidirectional bus, X[7:0], and routed to I/O pins. The registers each take four CLBs and should be arranged in columns. The soft macro counters also take four CLBs each. The part is an XC4003 with a 10 x 10 array of CLBs. Each quadrant is, therefore, a 5 x 5 array that can contain the logic.

Figure 15 shows a good placement that does not waste Longline resources. This placement constrains the bus and its multiplexed functions to one quadrant. It, therefore, only needs one set of horizontal and vertical Longlines. The bus I/Os are conveniently located to the left-hand end of the TBUF Longlines that drive them. A second choice for I/O placement would be at the right-hand end of the TBUF Longlines. This would, however, require the use of full-length TBUF Longlines.

Figure 16 shows an example of poor placement. It uses both halves of the horizontal Longlines to construct the 3-state bus, and additionally, might have to use both halves of the vertical Longlines to route the enable signals to the TBUFs. When the bus I/Os are located on the top or bottom of the graph (X6 and X7), routing them is difficult. X0-5 are not aligned with the bus, and require extra routing resources. This implementation would be slower than the previous one and consume more resources.

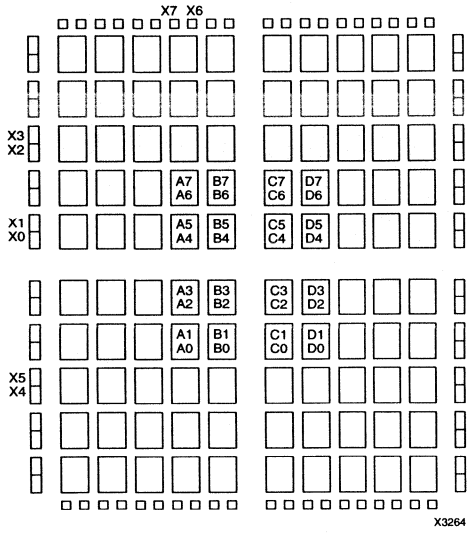


Figure 16. Inefficient Placement in an XC4003

Hard Macros

In the previous section, portions of the design were assigned fixed locations in the CLB array. If these fixed locations are chosen poorly, it can be more difficult for PPR to complete the design. There may, for instance, be many signals that must connect to the top of logic that is placed near the top of the array, thereby forcing PPR to generate the signals undesirably far from their destination.

This complication can be removed by creating Hard Macros. Hard Macros let the designer specify the placement of critical logic elements relative to each other, while allowing PPR the freedom to place the group where it is most convenient for the completion of the design. Unless specific I/O or Longline resources are needed, relative placement provides all the benefits of absolute placement without over-constraining PPR.

There are, however, factors that must be considered while creating Hard Macros. Firstly, all Hard Macros are rectangular; non-rectangular groups of CLBs convert to rectangular macros large enough to contain the original CLB group. In the process, empty CLBs are added that are not available to PPR, and will, therefore, remain unused in the final design. Holes in Hard Macros also remain unused. Leaving too many CLBs unused will cause a design to outgrow a device in which it would otherwise fit.

Secondly, the function generators in Hard Macros are closed, that is, no additional logic can be added into them. With a regular macro, an inverter driving a control pin will be absorbed into the macro, changing all the function generators to which the control pin connects but requiring no additional resources. With a Hard Macro, this is not possible and an additional function generator, outside of the Hard Macro, might be necessary just to implement the inverter.

The third factor is pin-swappability. PPR often swaps function generator pins to improve routability. However, when a Hard Macro includes a carry-logic function, all the pins are locked on CLBs that use carry logic. Since PPR can no longer swap pins to improve routability, pins must be carefully selected when the macro is created.

Hard Macros are created from an unrouted LCA design that contains just the desired CLBs. The .lca file is converted to a Hard Macro using the program HMGEN. Also available from the Xilinx Technical Hotline is a Hard Macro Style Guide document that describes how to create macros that are compatible with those provided in the Xilinx library.

Locking Down I/Os

PPR permits the user to lock I/O signals to specific pins. Pin locking is sometimes necessary to ease congestion on the printed-circuit board, match the pin-out of an existing socket, or simply to allow printed-circuit-board design to start before the FPGA design is complete. Like any other constraint, however, pin locking limits PPR, potentially reducing performance or even preventing the design from routing completely.

Any locking of I/Os should be done as late as possible in the design process. If possible, 75 - 80% of the design should be completed before I/Os are locked. At this stage, the preferred pin locations for the FPGA design will be known, and a workable compromise between the needs of the FPGA and the needs of the printed-circuit board can be reached.

While it might be convenient to lock pins earlier, there is a danger that the pin constraints will prevent the FPGA design from completing successfully. When this occurs, the only solution is to remove some of the pin constraints, thus invalidating any printed-circuit-board design that has already been done.

Software Techniques

XACT Performance

With XACT Performance, the designer can specify timing objectives in XC4000 designs. These objectives are used by PPR primarily to optimize its use of routing resources. If a critical net needs a particular routing resource that has already been allocated to a non-critical net, the timing-driven router can choose to re-route the non-critical net, thus making the desired resource available. Timing objectives permit PPR to make such trade-offs intelligently, by telling it which nets are critical and how much freedom it has to slow non-critical nets.

XACT Performance does:

- Provide easily understood control of critical routing
- Allow PPR more freedom in critical areas

- Compensate for logic depth wherever possible
- Give early warning if timing expectations are unrealistic

XACT Performance does not:

- Perform delay matching
- Permit arbitrarily deep logic
- Guarantee that timing specifications will be met

This section provides hints and suggests options that can be used with XACT Performance. For a full description of XACT Performance that defines the terminology used in this section, see the *XACT Development System Reference Guide.*

Effective use of XACT Performance requires an understanding of two fundamental concepts, the Forward Tracing mechanism, by which net attributes are applied to paths, and the arbitration mechanism used when multiple attributes apply to the same path.

Forward Tracing: Although TIMESPEC attributes are applied to nets, they specify path delays between flip-flops, or between flip-flops and I/O. Each TIMESPEC applies to a group of flip-flops that is determined by tracing forward from the net with the TIMESPEC attribute. The TIMESPEC is applied to any flip-flop with an input that can be reached from the TIMESPEC net either directly, or through any depth of combinatorial logic.

Multiple Attributes: In many cases, more than one TIMESPEC will trace forward to the same flip-flop. Such conflicts are resolved according to the pin type on which the TIMESPEC arrives at the flip-flop. The arrival pin priority is shown in Table 2. By selecting an appropriate net for the TIMESPEC attribute, TIMESPEC priority can be set in the schematic. If more than one TIMESPEC has the same priority, the fastest TIMESPEC wins. Separate arbitrations occur for C2S, P2S and C2P specifications.

Once flip-flops have been assigned TIMESPECs, TIMESPECs can be assigned to paths. The TIMESPEC for a path is the faster of the TIMESPECs at its source and destination. If either the source or destination flip-flop does not have a TIMESPEC, the path does not have a TIMESPEC. This situation can only occur if default TIMESPECs have been set to IGNORE.

As stated above, XACT Performance operates primarily by allocating routing resources according to a net criticality. The simplest tactic that gives critical nets maximum access to routing resources is to minimize the criticality of non-critical nets. PPR, however, calculates default specifications for paths with no user TIMESPECs, and sometimes these defaults are unnecessarily demanding. Critical paths can benefit if the objectives of non-critical paths are reduced to match the design requirement.

In light of the "fastest wins" rule, the unattached attributes, DC2S, DP2S and DC2P, should be set to the slowest requirement for each path type. Other faster paths can then be specified explicitly. Alternatively, if there are "don't care" paths, the unattached attributes can be set to IGNORE.

Another way PPR creates default specifications is by modifying a flip-flop C2S specification to provide missing P2S or C2P specifications. Again, these defaults can be unnecessarily demanding. This problem is solved by providing the missing specifications, or disabling the default mechanism by setting EXTEND_C2S = FALSE, as described in the PPR Options section.

When PPR creates default specifications for a design, it does so by estimating a typical path delay based on logic depth in the design. The same estimate can be used in any TIMESPEC by setting the delay to AUTO, rather than a number of nanoseconds or megahertz.

AUTO is beneficial when a realistic estimate of achievable delay is not available. Overly loose specifications lead to unnecessarily slow results, since PPR stops improving a design once it has met the specified objectives. Impossibly short delay specifications are equally unproductive. As the optimizer tries to perform the impossible task, it will often create excessively long path delays, and overall, the design will be slower that one using AUTO delays.

Table 2.

Specification Level	Specification Method on a Schematic	Comments
1	Unattached attribute of type DP2S, DC2S or DC2P (the leading D is used to indicate a default attribute.)	Applies to all flip-flops in the design. Can be overridden by a Level 2 or Level 3 specification.
2	TS attributes of type C2S, P2S, or C2P plus corresponding flag attached to a net that can be traced forward to flip-flop clock pins.	Applies to all flip-flops whose clock pins are reached by the forward tracing mechanism. Can be overridden by a Level 3 specification.
3	TS attribute of type C2S, P2S, or C2P plus corresponding TS flag attached to a net that can be traced forward to flip-flop input pins other than clock pins.	Applies to all flip-flops whose non-clock pins are reached by the forward tracing mechanism. Overrides a Level 1 or Level 2 specification.

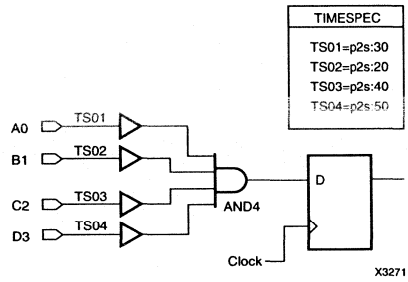


Figure 17. TIMESPEC Placement Where Some Specifications are Overridden

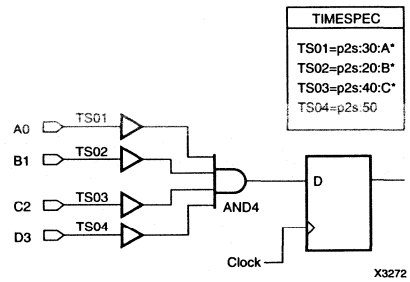


Figure 18. TIMESPEC Placement Where Some Specifications are Lost

Whenever AUTO is invoked in a TIMESPEC on a net, a typical path delay is calculated based on only the logic to which the particular TIMESPEC applies. Each automatic TIMESPEC creates a different specification for the logic it controls, and consequently, multiple AUTOs lead to different speed objectives for different parts of the design. Exploiting these differences can benefit designs containing both simple logic that must be fast, and complex logic that can be slower.

Automatic TIMESPECs should placed in the design such that each TIMESPEC controls logic with a particular complexity and speed requirement. This arrangement causes the objectives for simple logic to be faster than those for complex logic, as desired. A single automatic TIMESPEC for the whole design, would set the objective according to the most complex logic, and no attempt would be made to make the simple logic faster.

A frequent cause for concern is the XACT Performance warning message [pa:SPEC_DOES_NOT_APPLY]. This does not necessarily mean that a specification was wrongly entered or has been ignored by XACT. As stated above, multiple TIMESPECs are often traced to a single flip-flop, where only one can be applied. After eliminating the redundant specifications, the arbiter documents its activity by issuing the warning.

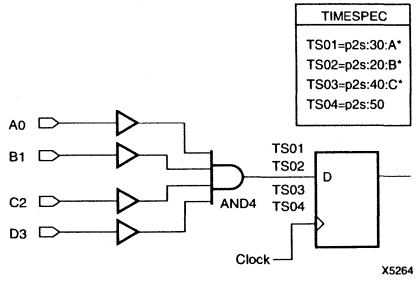


Figure 19. TIMESPEC Placement Where All Specifications are Retained and Apply

The number of warnings is increased if the same TIMESPEC is applied to many nets and several of them trace to a single flip-flop. Each copy that arrives at the flip-flop is treated as a separate specification, thus increasing the number of specifications that must be eliminated and the number of warnings that are created. Careful placement of TIMESPECs will minimize the number of warning messages.

There are times, however, when P2S or C2P specifications can be completely overridden or ignored. In Figure 17, for example, all four TIMESPECs trace to the flip-flop D-pin, and the fastest, TS02, wins. TS01, TS03 and TS04 are overridden. Figure 18 shows an even worse example. Again, TS02 wins, but it is limited to paths from pads with name that start with "B" (the wild-card group B*). Consequently, the paths from A0, C2 and D3 are treated as unspecified, in spite of their TIMESPECs.

The solution is shown in Figure 19. When multiple P2S TIMESPECs are placed on a single net and they are qualified to operate with separate pads, all of the specifications are retained if the net wins in the arbitration. All specifications are then used by XACT Performance, each specifying the paths from the group of pads for which it is qualified.

In the example, there is only one net with TIMESPECs attached, and so it must win. All four specifications are retained since they come from the winning net. TS01 is used for A0, since it is qualified for the A* group. Similarly, TS02 is used for B1 and TS03 is used for C2. TS04 is used for all pads not in the A*, B* or C* groups. In this case, only D3 remains to be controlled by TS04. To simplify the schematic, the four TIMESPECs could be linked into a single specification, TS05, and this single TIMESPEC attached to the net in place of the other four.

When creating custom hard macros for use with XACT Performance, following these guidelines will help PPR analyze the design more effectively.

- If there is a clock pin on the Hard Macro, make certain that it is named C. This allows PPR to recognize the presence of flip-flops in the macro. If there is no clock pin on the hard macro, no other pin should be named C.

- Avoid creating hard macros with more than one clock, since PPR can recognize only one clock input per hard macro.
- Avoid combining clocked and non-clocked outputs on the same hard macro. When PPR detects a clock pin on a hard macro, it assumes that all outputs of hard macro are sourced by flip-flops.

Prior to PPR v1.31, two problems existed when using Hard Macros with XACT Performance. First, PPR was unable to analyze Hard-Macro timing adequately during the partitioning and placement phases of the implementation. This would result in early PPR predictions that timing requirements could be met, that were later proven untrue during routing. Second, analysis of false paths through carry logic would result in false notification that timing requirements could not be met.

Both problems have been resolved in PPR v1.31. However, there are still some difference between PPR and XDELAY timing analyses. These are described in the section on XDELAY.

When analyzing clock-to-set-up paths, PPR ignores clock skew. This is appropriate if clocks are distributed on global nets using BUFGP or BUFGS buffers. If clocks are routed on local nets, however, the skew can be significant, and must be anticipated when setting TIMESPECs.

To ensure correct operation, all completed designs should be carefully analyzed using XDELAY, and this is especially true when clock skew may be present. Since the -Analyze option of XDELAY does not consider clock skew either, a separate clock analysis must be performed. If PPR and XDELAY timing results differ, the XDELAY results should be used.

PPR Options

As its name suggests, PPR performs three principal functions: partitioning, placement and routing. In the partitioning phase, schematic gates are collected into function generator-sized groups. These groups and any associated flip-flops are then assigned to specific CLBs in the placement phase. First, an initial placement is created, typically using a Mincut algorithm. This placement is then improved by analyzing connection distances. Routing is an iterative process where early connections are often re-routed in order to free critical resources for nets that are routed later.

This section describes the more important PPR options and their recommended settings. At the end of the section, two examples of PPR settings are given.

DC2P = {(number), "auto", "ignore"}
DC2S = {(number), "auto", "ignore"}
DP2S = {(number), "auto", "ignore"}
DP2P = {(number), "auto", "ignore"}

Entry Method: command line, paramfile or /pprdx2x in XACTINIT.DAT file

The parameters are character strings that represent XACT Performance Default-Clock-to-Pad (DC2P), Default-Clock-to-Set-up (DC2S), Default-Pad-to-Set-up (DP2S) and Default-Pad-to-Pad (DP2P) specifications. The path-delay objectives may be specified in tenths of nanoseconds (0.1 to 3000.0), "auto" for individual automatic setting by PPR, or "ignore" to request timing be ignored for particular path types (except where explicit specifications are defined in the schematic).

For example, if there are no explicit TIMESPECs on P2P paths, setting DP2P to "ignore" will cause PPR to ignore all P2P paths in the design. If the design has no P2P requirements, this setting helps PPR meet the TIMESPEC requirements that are present. If DP2P is set to "ignore," explicit TIMESPECs that deal with P2P paths are honored, but only paths with P2P TIMESPECs are optimized; the delay could increase on other important paths. The default setting for these parameters is "auto".

Recommended setting:

ignore	if unconcerned with unspecified paths.
(number)	if trying to reach a specific delay.
auto	to let PPR decide a reasonable delay to try for.

extend_c2s={True, False}

Entry Method: command line, paramfile or /ppr/extend_c2s in XACTINIT.DAT file

When the variable is True, PPR automatically generates a P2S and C2P TIMESPEC for each C2S TIMESPEC without a corresponding P2S or C2P. The specifications are assigned reasonable values. To ignore P2S and C2P paths unless they have explicit TIMESPECs, set the variable to False. PPR can then concentrate on the C2S TIMESPECs, without optimizing P2S and C2P paths unnecessarily. The default setting is True.

Recommended setting:

False	To avoid unnecessarily demanding. automatic specifications.
True	To extend C2S specifications to P2S and C2P paths.

ripup_allowance = (number)

Entry Method: /ppr/mxmazer/ripup_allowance = (number) in XACTINIT.DAT file

This option tells the router how many regressions to permit in the rip-up and retry process. Higher allowances increase the probability that a design will complete successfully, since the router tries longer. For example, if the router is close to its allowance and finds a minimum in the number of unroutes, it may stop. With a larger allowance, it might continue, and achieve a better result, although the number of unroutes may increase temporarily.

When running PPR multiple times to obtain the best placement, set the ripup_allowance low. Otherwise PPR can take a long time to produce a design that does not completely route anyhow due to a poor placement. By setting the parameter low, PPR can generate a result more quickly, and can go on to the next iteration of the loop.

If a design has hundreds of unroutes with a small ripup_allowance, increasing the allowance will probably not make it route completely. If the number of unroutes is 0-30, however, re-running PPR with a higher ripup_allowance will probably completely route the design. The default for this parameter is 2.

Recommended setting:

-1	Rips up and reroutes exhaustively, use if design will route or comes close to routing.
5-10	Use if unsure whether a design will route, or when running PPR loops.

improvecount = (number)

Entry Method: command line/paramfile

This option sets the number of iterations PPR makes to improve placement. The higher the number, the longer PPR attempts to improve the placement. If the placement ceases to improve, PPR stops with the best result it has obtained, regardless of the improvecount setting. Default = 3.

Recommended setting:

20	If runtime is not a factor.
6 - 8	Normally gives good results.

seeds_to_try = (number)

Entry Method: command line/paramfile

This option defines how many different seeds PPR tries in the placement improvement phase. For each of the specified number of seeds, PPR makes an initial placement and runs one improvement iteration. After this, the placement algorithm continues for (improvecount-1) iterations starting with the result that had the best score. Default = 1.

Recommended setting:

10	If runtime is not a factor.
3 - 4	Normally gives good results.

mincut_passes = (number)

Entry Method: command line/paramfile

This option sets the maximum improvement passes per partitioning attempt. Partitioning stops earlier if any pass shows no improvement. Default = 12.

Recommended setting:

20	If runtime is not a factor.
12	Reasonable value.

mincut_method = (number)

Entry Method: command line/paramfile

This variable selects the algorithm used during initial placement. The legal values are 0-3, and the default is method 3, which has proven to be best overall. On particular designs, however, methods 0-2 may yield better results. These other methods are most useful with designs that have low I/O utilization and/or large differences between flip-flop and function-generator utilization (check the PPR.LOG file for usage percentages).

Use these methods when placement achieved by method 3 is inadequate. There is no guarantee that the placement will improve, but a little experimentation can be very worthwhile. Default = 3.

Recommended setting:

3	Factory-tuned to normally yield good results.
0,1,2	Use experimentally to improve placement.

mincut_tries = (number)

Entry Method: command line/paramfile

This option sets the maximum number of initial configurations per partitioning step. While constructing a good initial partition, and improving upon it, a number of attempts are made at each partitioning step. The best result from these attempts is used in the design. Default = 2.

Recommended setting:

10	If runtime is not a factor.
2	For reasonable quality.

justflatten = (True, False)

Entry Method: command line/paramfile

When this option is set to True, the design is simply flattened into an .LCA file; the design is neither placed nor routed. Hard Macros, if they exist, are merged into the design. Using this option, a design is quickly translated into an .LCA file for back-annotation and unit delay simulation. This is only necessary when there are hard macros in the design, and unit-delay simulation is required. Default is False.

Recommendations

The following sets of PPR options are recommended. For better results than are given by the default settings at the expense of a slightly longer run-time, use these values.

On the command line or in the paramfile:

```
improvecount = 7
seeds_to_try = 3
extend_c2s = False
DC2P = ignore      (if not concerned
DP2S = ignore      with unspecified
DP2P = ignore      paths.)
```

In the XACTINIT.DAT file:

/ppr/mxmazer/ripup_allowance = 8

For results that are potentially much better than those given by the default settings but with a much longer runtime, use the following values.

On the command line or in the paramfile:

```
improvecount = 20
seeds_to_try = 10
extend_c2s = False
mincut_passes = 20
mincut_tries = 10
DC2P = ignore      (if not concerned
DP2S = ignore      with unspecified
DP2P = ignore      paths.)
```

In XACTINIT.DAT file:

ppr/mxmazer/ripup_allowance = -1

The XDELAY Static Timing Analyzer

XDELAY provides static timing analysis of Xilinx FPGA designs. It analyzes all logic paths through a design, and produces a report that can be automatically analyzed for worst-case performance, viewed on-line, or stored as a file. XDELAY is documented in the *XACT Development System Reference Guide*.

Since, the XDELAY report for a single path can require many lines of text, and large designs can contain thousands of paths, XDELAY report files can be very large, sometimes occupying several megabytes of disk space. The size of these reports can be reduced significantly by using the filters provided.

Reports can be restricted to a single path type, clock-to-set-up, for example, eliminating the reports on path types that are of no interest. For a worst-case-path analysis, setting the Delaygreater variable can eliminate paths that are too fast to be relevant. Localized analysis can be performed using the From, To or Ignorenet filters to limit the scope of the search.

PPR also provides static timing analysis as a part of its report, and sometimes there are differences between XDELAY and PPR timing results. These difference arise either because PPR does not trace certain paths, or because is analyzes some delays incorrectly.

XDELAY is the definitive Xilinx-FPGA timing analyzer, and its results should have more authority than those from any other source. XDELAY analysis should be part of the design process for all designs.

The following situations can result in PPR tracing paths differently than XDELAY.

- PPR does not trace through the asynchronous path from a reset-direct or set-direct pin to a flip-flop output. By default, however, XDELAY traces through this asynchronous path. The XDELAY option Flagblk CLB_Disable_SR_Q disables this tracing. See the item *PPR v1.30: Asynchronous Set/Reset Inputs Treated as Path Endpoints* in the release notes for more information.
- PPR treats RAM elements as combinatorial logic and continues tracing to the next clocked element or I/O pin. See the item *PPR V1.30: RAM Elements are NOT treated as Path Endpoints* in the release notes for more information.
- PPR does not trace through bi-directional IOBs as XDELAY sometimes does. See the item *XDELAY v4.30: Flagblk IOB_Disable_O_I and IOB_Disable_T_I Explained* in the release notes for more information.

The situations that causes PPR to analyze delays incorrectly are as follows.

- PPR always uses T_{IO1} as the delay through a TBUF. T_{IO1} is the delay from the TBUF I-pin to the output. Consequently, the delay from the T-pin of the TBUF will be incorrectly assigned the T_{IO1} delay. In some cases, T_{IO1} is less than the correct delay. WAND and WORAND elements are implemented with TBUFs, and are, therefore, also affected by this problem.
- If a BUFGP is sourced by internal logic, PPR must route that signal out through the dedicated IOB and back into the BUFGP input. In computing the delay of this path, PPR assumes that the output driver is configured for the FAST slew-rate, which is usually not the case. Thus, the delay may under-reported.
- Small discrepancies, typically less than 1 ns, appear in various places due to modeling differences between PPR and XDELAY/LCA2XNF.

LCA Speed Estimation: Asking the Right Question

XAPP 011.001

Application Note By BERNIE NEW

Summary

A simple algorithm is described for determining the depth of logic, in CLBs, that can be supported at a given clock frequency. The algorithm is suitable for XC3000/XC3100 or XC4000 LCA devices.

Speed is always a consideration when deciding whether a design can be implemented in an LCA devices. Often, an initial logic design is created and the question asked, "How fast will this run in an LCA device?"

This is not an easy question to answer. A good speed estimate requires careful analysis of the logic design; performance will vary with the logic implementation. To complicate matters, routing delays are always unknown at this stage.

When the estimate is complete, it is usually compared to a given system requirement simply to determine adequacy, and the exact number becomes irrelevant. If a system requires 30 MHz, for example, being able to operate at 35, 40 or even 50 MHz makes no difference.

A better question is "Will an LCA implementation meet the system speed requirements?"

This can often be answered much more easily. Given a required clock rate, it is easy to estimate the level of complexity that can be supported. This complexity can then be compared to the functional requirements to make an initial determination of feasibility. Only in marginal cases does a full speed estimate become necessary.

A typical data path runs from a register, through some combinatorial logic to another register. In an LCA device, this requires, as a minimum, a CLB clock-to-output delay plus a set-up time. In an XC3000-125 part, these total 10.5 ns. Including routing, 15 ns should be typically allowed. If combinatorial CLBs are added into the path, each level of CLBs adds 5.5 ns. Additional routing delays are also created. Including a typical routing allowance, 10 ns should be added for each level of combinatorial CLBs.

This simple speed-estimating procedure can also be reversed. If, for example, the system clock frequency is 30 MHz, the 33 ns period typically provides for two combinatorial CLBs.

Clock period	33 ns
Minimum delay	-15 ns
	18 ns
Combinatorial delay	÷10 ns
	~2 CLBs

Including the function generator in the destination CLB, a total of three function generators can be cascaded. If the number of function generators that can be cascaded is known, the design can be analyzed to determine whether or not it is feasible.

This should not be considered a hard limit. Shorter routing delays can be achieved, allowing deeper logic. However, dependence on short routing delays will probably necessitate optimization of both the logic design and the routing.

Nor is the number of function generators guaranteed. Longer routing delays may be encountered, especially if a chip is fully utilized or if high fan-out signals are used. Elimination of these long routing delays may necessitate manual routing or logic design changes. In any case, the timing of all LCA designs should be analysed after routing to determine worst-case performance.

Table 1 shows typical minimum delays for various LCA devices. Also shown are typical increments for combinatorial CLBs. To allow for higher routing delays, these figures should be increased by 5 ns, if more that 60 – 75% of the CLBs are to be used. If a large LCA device is to be used and the CLBs are placed automatically, a separate 3 – 5 ns should be added to each delay.

This technique not only simplifies the feasibility study, it also provides valuable information on which to base the logic design. Critical areas can be identified prior to starting the design. It is better to design around the critical areas than to have to accommodate them during implementation. Conversely, if a design only requires a fraction of the capability available, it might be possible to multiplex some functions to provide a less costly implementation.

Table 1. Delays, Including Typical Routing

	XC3000			XC3100			XC4000		
	-70	-100	-125	-5	-4	-3	-6	-5	
Minimum delay	21	18	15	10	9	7	17	12	ns
Combinational delay	15	12	10	8	7	6	12	9	ns

To each delay add: 5 ns for high utilization
3 – 5 ns for large LCA Devices

Using the XC4000 Readback Capability

XAPP 015.000

Application Note By WOLFGANG HÖFLICH

Summary

This Application Note describes the XC4000 Readback capability and its use. Topics include: initialization of the Readback feature, format of the configuration and Readback bitstreams, timing considerations, software support for reading back LCA devices, and Cyclic Redundancy Check (CRC).

Xilinx Family	*Demonstrates*
XC4000	XC4000 Readback Capability

Purpose

Every LCA device shipped by Xilinx is tested using the device Readback capability. All CLBs and IOBs are configured and read back using extensive test patterns to guarantee 100% functionality of the LCA device.

An LCA device can be read back at any time after configuration.The Readback data consists of the configuration data and, optionally, the current state of the CLBs and IOBs.

When is a Readback Necessary or Useful?

The Xilinx devices are 100% pretested and the XC4000 series LCA devices can use Cyclic Redundancy Checking (CRC) on the configuration bitstream to check the integrity of the bitstream loaded into the LCA configuration memory.

In the configuration bitstream, there are four error-check bits for each data frame transmitted into the LCA device. Using this technique, the LCA device detects invalid data bits and aborts the configuration process. The INIT status pin is pulled Low, signaling that an error occurred during loading of the configuration memory.

Therefore, Readback is useful only in few cases.

- Verifying the configuration in a very unstable environment,
- Reading back the internal state of the RAM, CLBs and IOBs during the LCA development phase,
- In high-reliability applications that require in-system functional analysis and verification,
- For Xilinx internal testing

For examples of how to use Readback in your application, contact Xilinx.

Readback Highlights

The Readback features and the user interface of the XC4000 devices are significantly improved over the XC2000/XC3000 devices.

The Readback operation does not interfere with the LCA operation. After a valid Readback request, the current state of LCA internal nodes can be captured into a special shift register. Then the data can be transferred out of the device using a user-defined clock signal.

The following LCA internal configuration data and circuit nodes are available for Readback (Figure 1).

- Configuration memory bits that define the logic configration of CLBs, IOBs, and the LCA interconnects.
- X and Y output pins of CLB Function Generators.
- XQ and YQ output pins of CLB flip-flops,
- OQ output pins of IOB flip-flops,
- I1 and I2 input pins of IOBs

A mask file (<design_name>.LL), generated with the MakeBits program, contains information about the location of the user data bits in the Readback bitstream and the names of the signals connected.

The user can implement comparison logic in CLBs to perform the comparison with data stored in the configuration PROM. This technique does not work if any CLB is used as RAM, since changing the RAM contents alters the data in the configuration memory. In this case, an additional mask PROM is needed to disable the comparison of Readback bitstream locations that represent the RAM data.

The Readback speed is 10 kHz min, 1 MHz max. See the timing diagrams at the end of this application note.

The XC4000 family features a Boundary-Scan instruction that initiates a Readback sequence using the standard IEEE 1149.l/JTAG Boundary-Scan ports.

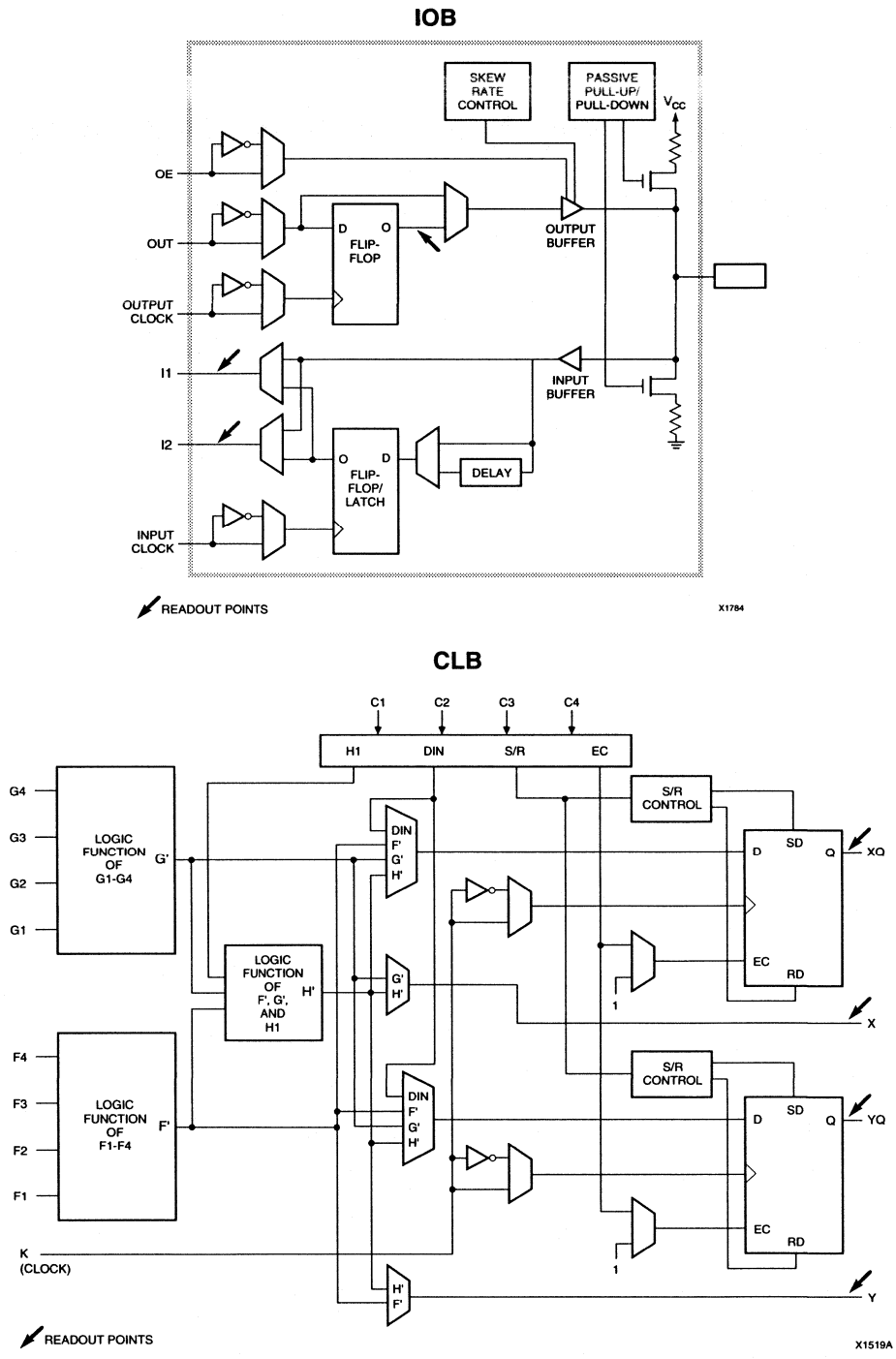


Figure 1. Readback Capture Enable

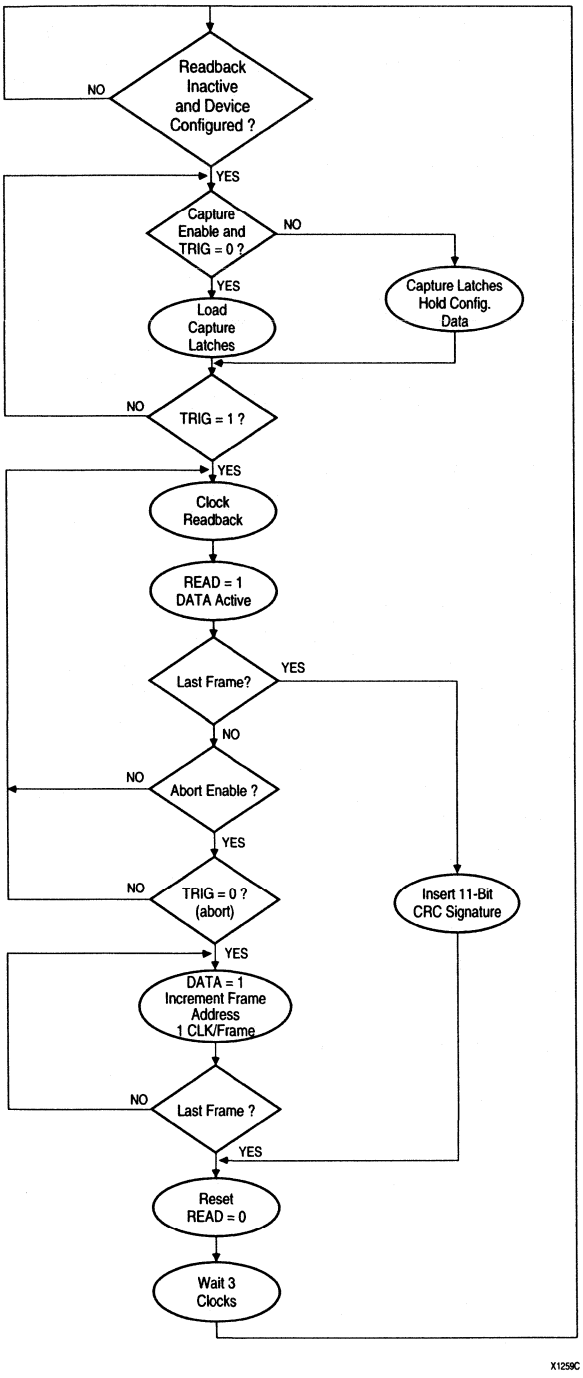


Figure 2. Readback State Diagram

Daisy chaining LCA devices for Readback is not possible. Each device must be read back individually.

The XChecker Universal Download Cable and Logic Probe handles configuration and Readback of XC2000, XC3000, and XC4000 FPGA families. In addition, it displays selected LCA internal nodes on screen.

Performing a Readback

Readback State Diagram

An LCA-internal state machine controls the Readback process. See Figure 2 for the Readback state diagram. For an explanation of the terms used, see below.

Readback Primitive

The XC4000 LCA device has a dedicated primitive that handles all of the Readback functions. It is located in the lower left and right corners of the LCA device and has two inputs and two outputs (Figure 3).

The Readback primitive can access general-purpose interconnects. Therefore, the four signals – rdclk.I, rdbk.TRIG, rdbk.RIP, and rdbk.DATA – can connect to the user I/Os and to CLBs as follows.

- rdclk.I – The Clock input can be connected to any device input pin, or any CLB output. If it is not connected to a user net, it connects to the device CCLK input pin, if the appropriate option is selected in the bitstream-generator MakeBits program.

- rdbk.TRIG – A Low-to-High transition on the TRIG input starts a Readback sequence. The minimum required pulse width is one rdclk.I cycle. A valid trigger causes the current value of certain nodes to be latched into an LCA internal holding register. If ReadAbort was selected as an option in MakeBits, a High-to-Low on the TRIG input aborts the Readback. In this case, additional clocks must be provided until rdbk.RIP signals the end of a Readback. The rdbk.TRIG cannot be reasserted until at least three clock periods after the previous Readback has been terminated correctly.

- rdbk.RIP (Readback-In-Progress) – A High on this output indicates that a Readback is being performed. RIP goes active one Readback clock cycle after a valid Readback trigger has occurred. It goes Low with the last data

Figure 3. The Readback Primitive

bit shifted out of the LCA device. In the case of a Readback abort, RIP remains active until the Readback sequence is terminated correctly.

- rdbk.DATA – The Readback data is available on the DATA output of the Readback primitive. Each rising edge on rdclk.I shifts one data bit from the LCA-internal holding register to the DATA output. The data bitstream is explained below. There is an option to disable the user data bits in the Readback bitstream.

Note that in XC3000 devices, the input pin M0/RTRIG is used as a Readback Trigger pin and M1/RDATA as a Readback Data pin. In XC4000, the M0 pin can be used as an input pin, the M1 pin as a 3-state output.

Also, XC3000 has a MakeBits option to inhibit Readback. In XC4000, conventional Readback is possible if the Readback primitive is used in the design, or if a Boundary-Scan Readback is performed.

Readback Initialization

There are three ways of preparing an LCA design for Readback.

- Using the Readback primitive on the schematic.
- Activating Readback from the XACT Design Editor.
- Performing a Readback during a Boundary-Scan operation.

Readback from the schematic level
In the Xilinx Design Interface Libraries, there is a Readback primitive that can be called up into the schematic like any other library primitive. Simply connect the inputs and outputs of the Readback primitive to your user nets as desired. See Figure 4 for an example.

Note: If the CLK input is not connected to any net, the Place-and-Route software connects it to the CCLK input pin, if the appropriate ClkSelect=Cclk was selected in the MakeBits program.

Readback from the XDE
In XDE, the Readback primitive is located in the lower left and lower right corners of the device. It is activated if the

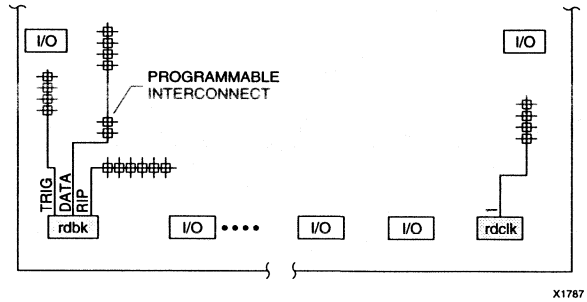


Figure 5. The XACT Readback Primitive

rdbk.TRIG and the rdbk.DATA signals are connected. The rdclk.I pin is connected to the CCLK pin, if not connected otherwise. See Figure 5.

Readback during a Boundary-Scan
No changes are required to prepare a design for Readback through the Boundary-Scan port. Contact Xilinx for additional information.

Configuration and Readback Bitstreams

The XC4000 Configuration Bitstream

Figure 6 shows the format of the XC4000 configuration bitstream, as generated by the XACT MakeBits program. The bitstream consists of header and program data. The header consists of four dummy bits, the preamble code, the configuration-program-length count, and an additional four dummy bits. The program data is divided into frames consisting of a Start bit (0), the data field, and four error check bits (eeee). The bitstream ends with eight or more postamble bits (01111XXX). The exact number of the bits in the bitstream is determined by the 24-bit program-length count.

The XC4000 Readback Bitstream

The Readback bitstream contains configuration information as well as the state of internal user logic. The Readback bitstream starts with five dummy bits. The Readback data frame has the same format as the configuration data frame which eases a bit-by-bit comparison between

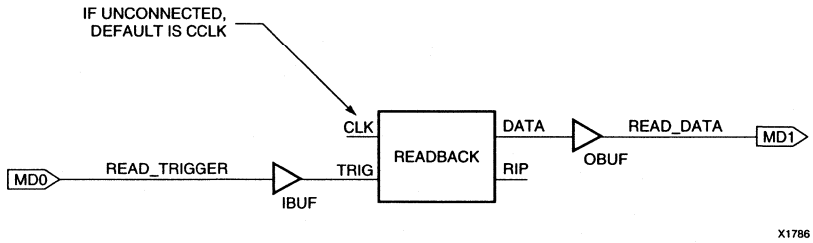


Figure 4. Readback Symbol on the Design Schematic

Readback and configuration data. Each data frame consists of a Start bit (0), the Data field, and four Stop bits (1111). The bitstream ends with 11 CRC bits, Figure 7.

Both the configuration data and the internal-logic data are included in the Readback bitstream. In the Readback bitstream, the configuration data bits are not inverted with respect to the configuration bitstream. The user-logic data bits, however, are inverted with respect to their values during Readback capture.

The read-back configuration data may differ from the original data downloaded into the device if CLB RAM is used in the design. The RAM data is stored in the F- and G-function tables of the CLB.

The first two bits of the first Readback data frame are variable; they are non-user, non-configuration bits. Their input state is dependent on the configuration speed and the configuration error-check mode of the LCA device. The last seven bits of the last Readback data frame are always ones.

If Readback capture of user data is disabled in the MakeBits program, logic Highs replace the user data. Note that the RAM data is not part of the captured user logic data; it is contained in the read-back configuration data.

The bitstream ends with eleven bits of a CRC signature appended. If ReadCapture is disabled and the design does not use any CLB RAM, this signature will be constant in successive Readbacks. See below for more information on the Polynomial Cyclic Redundancy Check CRC-16.

Software Support for Readback

The user can set Readback options with the MakeBits program.The following MakeBits options are relevant for Readback of XC4000 devices.

ReadCapture:

Settings: Enable, Disable
Default: Disable

This option determines whether the state of internal user logic is included in the Readback bitstream. If ReadCapture is disabled, the user data is replaced by ones.

ReadAbort:

Settings: Enable, Disable
Default: Disable

ReadAbort enables the level-sensitive signal rdbk.TRIG to abort the Readback. A High-to-Low transition stops the Readback. Additional clocks must be supplied to terminate the Readback correctly. As a minimum, the number of data frames contained in the device plus three must be sent as additional clocks. During this period, the Readback data is High. The rdbk.RIP signal indicates the completion of a Readback process.

ClkSelect:

Settings: CCLK, RDBK (user supplied)
Default: CCLK

The rdclk.I pin can be connected to any user net or to the CCLK I/O pin. With this option, the user can choose between the alternatives.

MakeBits features an option used to create a "logic allocation" file (<design_name>.LL) that contains information on which bit in the Readback bitstream corresponds to which signal in the design. This ASCII mask file indicates the offset from the beginning of the Readback bitstream, the frame number, the offset within a frame, and names of user signals in the Readback bitstream. Figure 8 shows an example.

Readback Timing

Minimum Readback frequency is 10 kHz; maximum Readback frequency is 1 MHz. The rdclk.I High time and Low time are each 0.5 µs min. See Table 1 for additional preliminary Readback switching characteristics.

Cyclic Redundancy Check (CRC) for LCA Configuration and Readback

Concept of the Cyclic Redundancy Check

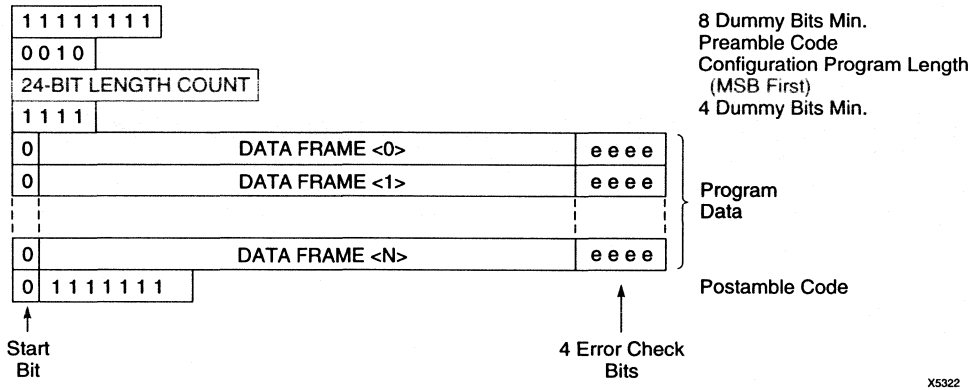
The Cyclic Redundancy Check is a method of error detection in data transmission applications. Generally, the transmitting system performs a calculation on the serial bitstream. The result of this calculation is tagged onto the data stream as additional check bits. The receiving system performs an identical calculation on the bitstream and compares the result with the received checksum. CRC Checksum Compare is often referred to as Signature Analysis.

CRC During LCA Configuration

Each data frame of the LCA configuration bitstream has four error bits at the end. See Figure 6. If a frame data error is detected during the loading of the LCA device, the configuration process with a potentially corrupted bitstream is terminated. The LCA pulls the INIT pin Low and goes into a Wait state.

CRC During LCA Readback

During an LCA Readback, 11 bits of the 16-bit checksum are appended to the end of the Readback data stream. The checksum is computed using the CRC-16 CCITT polynomial (Figure 9).The LCA checksum consists of the 11 most significant bits of the 16-bit code. A change in the checksum indicates a change in the Readback bitstream. Statistically, one in 2048 errors might go undetected.



Device	XC4002A	XC4003A	XC4003/3H	XC4004A	XC4005A	XC4005/5H	XC4006	XC4008	XC4010/10D	XC4025
Gates	2,000	3,000	3,000	4,000	5,000	5,000	6,000	8,000	10,000	20,000
CLBs	64	100	100	144	196	196	256	324	400	1,024
(Row x Col)	(8 X 8)	(10 X 10)	(10 X 10)	(12 X 12)	(14 X 14)	(14 X 14)	(16 X 16)	(18 X 18)	(20 X 20)	(32 X 32)
IOBs	64	80	80/160	96	112	112 (192)	128	144	160	256
Flip-Flops	256	360	360/300	480	616	616 (392)	768	936	1,120	2,560
Horizontal TBUF Longlines	16	20	20	24	28	28	32	36	40	64
TBUFs/Longlines	10	12	12	14	16	16	18	20	22	34
Bits per Frame	102	122	126	142	162	166	186	206	226	346
Frames	310	374	428	435	502	572	644	716	788	1,220
Program Data	31,628	45,636	53,936	62,204	81,332	94,960	119,792	147,504	178,096	422,128
PROMs size (bits)	31,668	45,676	53,976	62,244	81,372	95,000	119,832	147,544	178,136	422,168

XC4000, 4000H: Bits per Frame = (10 x number of Rows) + 7 for the top + 13 for the bottom + 1 + 1 start bit + 4 error check bits
Number of Frames = (36 x number of Columns) + 26 for the left edge + 41 for the right edge + 1

XC4000A: Bits per Frame = (10 x number of Rows) + 6 for the top + 10 for the bottom + 1 + 1 start bit + 4 error check bits
Number of Frames = (32 x number of Columns) + 21 for the left edge + 32 for the right edge + 1

Program Data = (Bits per Frame x Number of Frames) + 8 postamble bits

PROM Size = Program Data + 40

The user can add more "one" bits as leading dummy bits in the header, or, if CRC = off, as trailing dummy bits at the end of any frame, following the four error check bits, but the Length Count value **must** be adjusted for all such extra "one" bits, even for leading extra ones at the beginning of the header.

Figure 6. XC4000 Configuration Bitstream Format

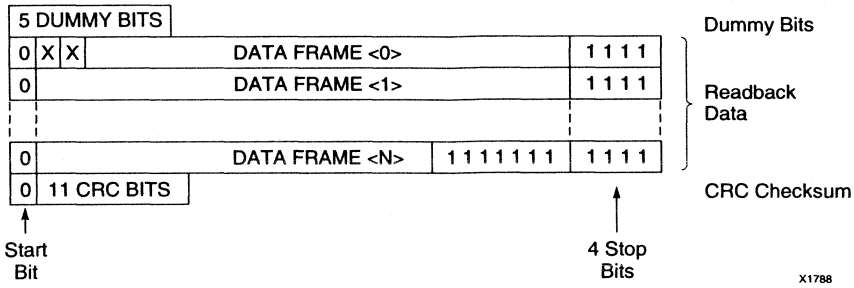
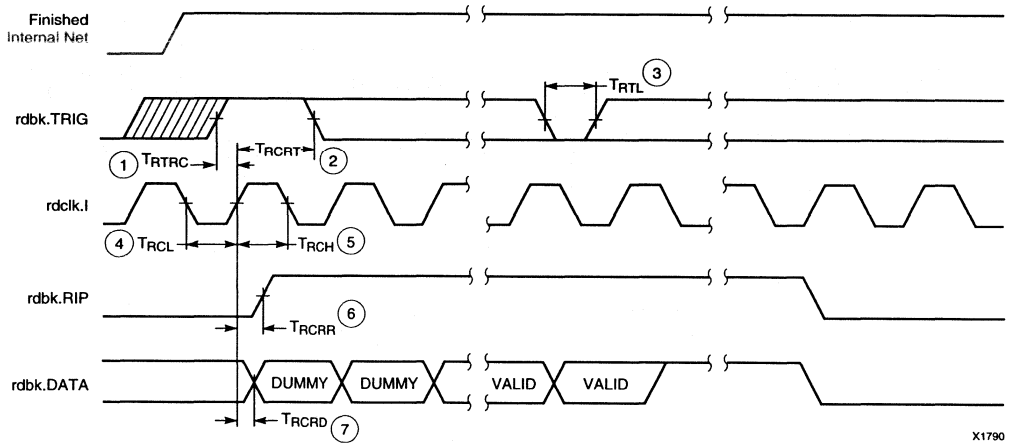


Figure 7. XC4000 Readback Bitstream

; Offset	Column (Frame)	Row (Frame Offset)	Description
21	1	100	P57 I1
32	1	90	U37 I1
41	1	79	P60 U1
.	.	.	.
.	.	.	.
.	.	.	.
36640	303	23	CD YQ
36650	303	13	BD YQ
37044	307	103	LD XQ CFG/TOGGLE
37054	307	93	KD XQ CFG/RDATA_REG/Q9
37064	307	83	JD XQ CFG/RDATA_REG/Q1
37074	307	73	ID XQ CFG/RDATA_REG/Q2
37084	307	63	HD XQ REFDATA_REG/Q5
37095	307	52	FD XQ
37105	307	42	ED XQ
.	.	.	.
.	.	.	.
.	.	.	.

Figure 8. Sample Logic Allocation File

Table 1. Readback Switching Characteristics



	Description	Symbol		Limits Min	Max	Units
rdbk.TRIG	rdbk.TRIG setup	1	T_{RTRC}	200	–	ns
	rdbk.TRIG hold	2	T_{RCRT}	50	–	ns
	rdbk.TRIG Low to abort Readback	3	T_{RTL}	100	–	ns
rdclk.I	rdbk.DATA delay	7	T_{RCRD}	–	250	ns
	rdbk.RIP delay	6	T_{RCRR}	–	250	ns
	High time	5	T_{RCH}	0.5	50	µs
	Low time	4	T_{RCL}	0.5	50	µs

Notes:
1. Timing parameters apply to all speed grades.
2. If rdbk.TRIG is High prior to Finished, Finished will trigger the first Readback.

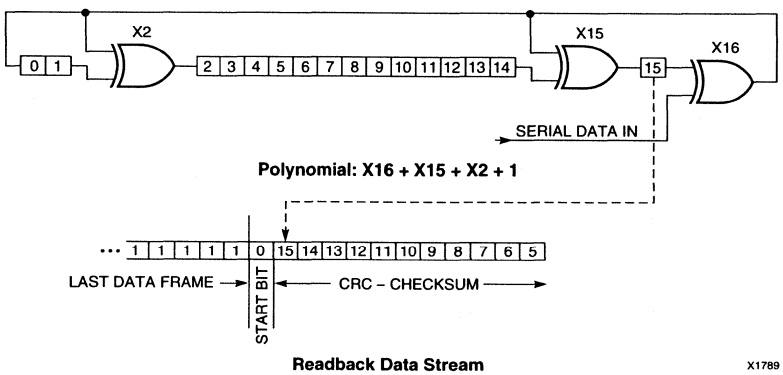


Figure 9. Circuit for Generating the CRC-16

Boundary Scan in XC4000 Devices

XAPP 017.002

Application Note By LUIS MORALES

Summary

XC4000 LCA devices contain boundary-scan facilities that are compatible with IEEE Standard 1149.1. This Application Note describes those facilities in detail, and explains how boundary scan is incorporated into an LCA design.

Xilinx Family	*Demonstrates*
XC4000	Boundary Scan

Introduction

In production, boards must be tested to assure the integrity of the components and the interconnections. However, as integrated circuits have become more complex and multi-layer PC-boards have become more dense, it has become increasingly difficult to test assembled boards.

Originally, manufacturers used functional tests, applying input stimuli to the input connectors of the board, and observing the results at the output. Later, "bed-of-nails" testing became popular, where a customized fixture presses sharp, nail-like stimulus- and test-probes into the exposed traces on the board. These probes were used to force signals onto the traces and observe the response.

However, increasingly dense multi-layer PC boards with ICs surface-mounted on both sides have stretched the capability of bed-of-nail testing to its limit, and the industry is forced to look for a better solution. Boundary-scan techniques provide that solution.

The inclusion of boundary-scan registers in ICs greatly improves the testability of boards. Boundary scan provides a mechanism for testing component I/Os and interconnections, while requiring as few as four additional pins and a minimum of additional logic in each IC. Component testing may also be supported in ICs with self-test capability.

Devices containing boundary scan have the capability of driving or observing the logic levels on I/O pins. To test the external interconnect, devices drive values onto their outputs and observe input values received from other devices. A central test controller compares the received data with expected results. Data to be driven onto outputs is distributed through a chain of shift registers, and observed input data is returned through the same shift-register path.

Data is passed serially from one device to the next, thus forming a boundary-scan path or loop that originates at the test controller and returns there. Any device can be temporarily removed from the boundary-scan path by bypassing its internal shift registers, and passing the serial data directly to the next device.

XC4000 LCA devices contain boundary-scan registers that are compatible with the IEEE Standard 1149.1, that was derived from a proposal by the Joint Test Action Group (JTAG). External (I/O and interconnect) testing is supported; there is also limited support for internal self-test.

Overview of XC4000 Boundary-Scan Features

XC4000 devices support all the mandatory boundary-scan instructions specified in the IEEE Standard 1149.1. A Test Access Port (TAP) and registers are provided that implement the EXTEST, SAMPLE/PRELOAD and BYPASS instructions. The TAP can also support two USERCODE instructions.

Boundary-scan operation is independent of individual IOB configuration and package type. All IOBs are treated as independently controlled bidirectional pins, including any unbonded IOBs. Retaining the bidirectional test capability even after configuration affords tremendous flexibility for interconnect testing.

Additionally, internal signals can be captured during EXTEST by connecting them to unbonded IOBs, or to the unused outputs in IOBs used as unidirectional input pins. This partially compensates for the lack of INTEST support.

The public boundary-scan instructions are always available prior to configuration. After configuration, the public instructions and any USERCODE instructions are only available if specified in the design. While SAMPLE and BYPASS are available during configuration, it is recommended that boundary-scan operations not be performed during this transitory period.

In addition to the test instructions outlined above, the boundary-scan circuitry can also be used to configure the LCA device, and read back the configuration data.

The following description assumes that the reader is familiar with boundary-scan testing and the IEEE Standard. Only issues specific to the XC4000 implementation are discussed in detail. For general information on boundary scan, please refer to the bibliography.

Deviations from the IEEE Standard

The XC4000 boundary scan implementation deviates from the IEEE standard in that three dedicated pins (CCLK, $\overline{\text{PROGRAM}}$ and DONE) are not scanned.

It should also be noted that the Test Data Register contains three Xilinx test bits (BSCANT.UPD, TDO.O and TDO.T) and that bits of the register may correspond to unbonded or unused pins.

Additionally, the EXTEST instruction incorporates INTEST-like functionality that is not specified in the standard, and system clock inputs are not disabled during EXTEST, as recommended in the standard.

The TAP pins (TMS, TCK, TDI and TDO) are scanned, but connections to the TAP controller are made before the boundary-scan logic. Consequently, the operation of the TAP controller cannot be affected by boundary-scan test data.

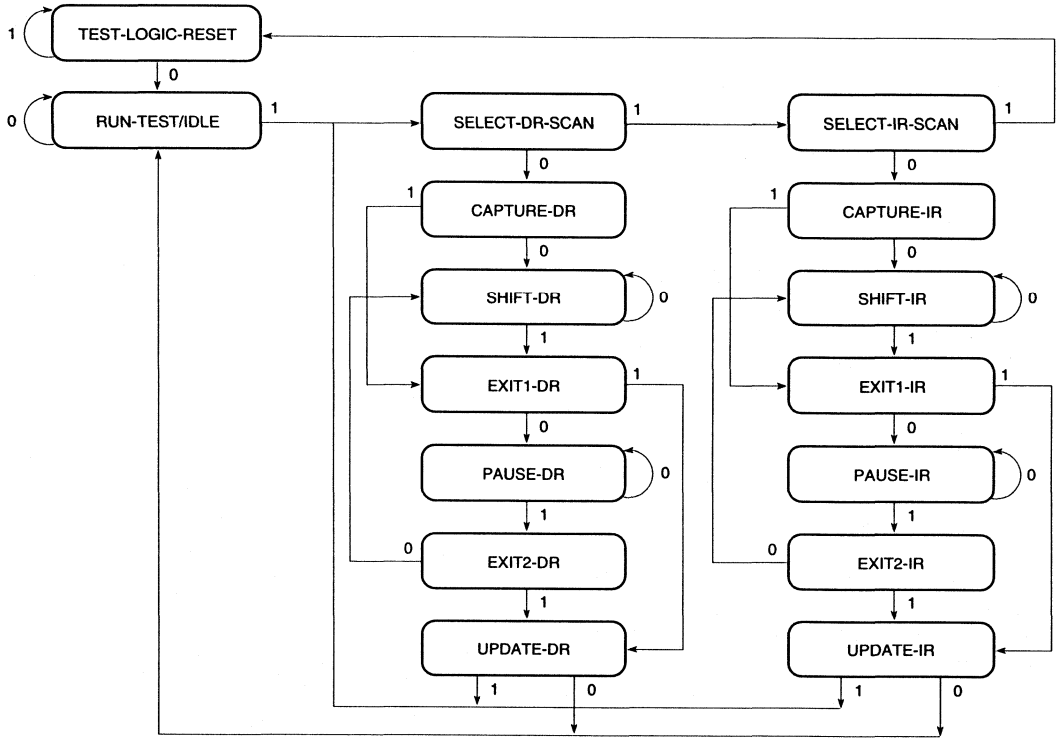
Boundary-Scan Hardware Description

Test Access Port

The boundary-scan logic is accessed through the Test Access Port, which comprises four semi-dedicated pins: Test Mode Select (TMS), Test Clock (TCK), Test Data Input (TDI) and Test Data Output (TDO), as defined in the IEEE specification.

The TAP pins are permanently connected to the boundary-scan circuitry. However, once the device is configured, the connections may be ignored unless the use of boundary scan is specified in the design (See "Using Boundary Scan").

If the use of boundary scan is specified, the TAP input pins (TMS, TCK and TDI) may still be shared with other logic, subject to limitations imposed by external connections and the operation of the TAP Controller. In designs



NOTE: The value shown adjacent to each state transition in this figure represents the signal present at TMS at the time of a rising edge at TCK.

X2680

Figure 1. State Diagram for the TAP Controller

that do not use boundary scan after configuration, the TAP pins can be used as inputs to or outputs from the user logic in the LCA device. TMS, TCK and TDI are available as unrestricted I/Os, while TDO only provides a 3-state output.

TAP Controller

The TAP Controller is a 16-state state machine that controls the operation of the boundary-scan circuitry in response to TMS. This state machine implements the state diagram specified by the IEEE standard, Figure 1, and is clocked by TCK.

Upon power-up or assertion of $\overline{\text{PROGRAM}}$, the TAP controller is forced into the Test-Logic-Reset state. After configuration, the controller is disabled, unless its use is explicitly specified in the user design.

Instruction Register

Loading a 3-bit instruction into the Instruction Register (IR) determines the subsequent operation of the boundary-scan logic, Table 1. The instruction selects the source of the TDO pin, and selects the source of device input and output data (boundary-scan register or input pin/user logic)

Note: In XC4000, whenever the TAP Controller is in the Shift-DR state, all data registers are shifted, regardless of the instruction. DR data is modified even if a BYPASS instruction is executed.

A 3-bit status word returned to the central test controller during an IR cycle comprises a boundary-scan availability flag, preceded by two mandatory bits; I0 is a one and I1 is a zero. This flag is High before and after configuration, when the full boundary-scan capability is available, and Low during configuration, when only SAMPLE/PRELOAD and BYPASS are available.

Table 1. Boundary Scan Instructions.

Instruction I_2	I_1	I_0	Test Selected	TDO Source	I/O Data Source
0	0	0	EXTEST	DR	DR
0	0	1	SAMPLE/ PRELOAD	DR	Pin/Logic
0	1	0	USER 1	TDO1	Pin/Logic
0	1	1	USER 2	TDO2	Pin/Logic
1	0	0	READBACK	Readback Data	Pin/Logic
1	0	1	CONFIGURE	DOUT	Disabled
1	1	0	RESERVED	—	—
1	1	1	BYPASS	Bypass Reg	Pin/Logic

I_0 is closest to DTO

The Boundary-Scan Data Register

The Data Register (DR) is a serial shift register implemented in the IOBs of the LCA device, Figure 2. Potentially, each IOB can be configured as an independently controlled bidirectional pin. Therefore, three data register bits are

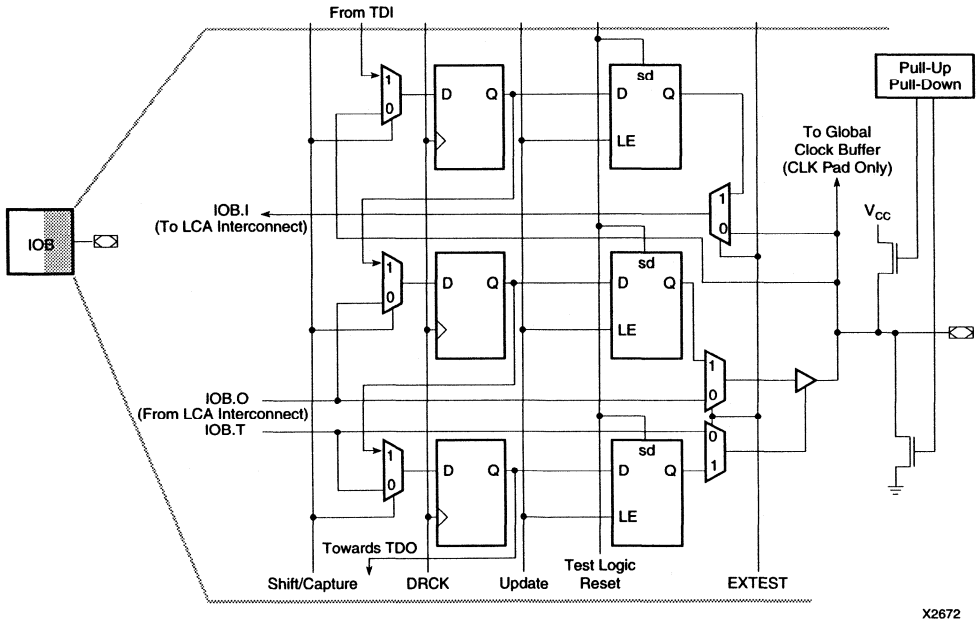


Figure 2. Boundary Scan Logic in a Typical IOB

provided per IOB: for input data, output data and 3-state control. In practice, many of these bits are redundant, but they are not removed from the scan chain.

An update latch accompanies each bit of the DR, that is used to hold injected test data stable during shifting. The update latch is opened during the Update-DR state of the TAP Controller when TCK is Low.

In a typical DR instruction, the DR captures data during the Capture-DR state (on the rising edge of TCK). This data is then shifted out and replaced with new test data. Subsequently, the update latch opens, and the new test data becomes available for injection into the logic or the interconnect. The injection of data occurs only if an EXTEST instruction is in progress.

Note: The update latch is opened whenever the TAP Controller is in the Update-DR state, regardless of the instruction. Care must be exercised to ensure that appropriate data is contained in the update latch prior to initiating an EXTEST. Any DR instruction, including BYPASS, that is executed after the test data is loaded, but before the EXTEST commences, changes the test data.

The IEEE Standard does not require the ability to inject data into the on-chip system logic and observe the results during EXTEST. However, this capability helps compensate for the lack of INTEST. Logic inputs may be set to specific levels by a SAMPLE/PRELOAD or EXTEST instruction and the resulting logic outputs captured during a subsequent EXTEST. It must be recognized, however, that all DR bits are captured during an EXTEST and, therefore, may change.

Pull-up and pull-down resistors remain active during boundary scan. Before and during configuration, all pins are pulled up. After configuration, the IOB can be configured with a pull-up resistor, a pull-down resistor or neither. Internal pull-up/pull-down resistors must be taken into account when designing test vectors to detect open circuit PC traces.

The primary and secondary global clock inputs (PGCK1-4 and SGCK1-4) are taken directly from the pins, and cannot be overwritten with boundary-scan data. However, if necessary, it is possible to drive the clock input from boundary scan. The external clock source is 3-stated, and the clock net is driven with boundary scan data through the output driver in the clock-pad IOB. If the clock-pad IOBs are used for non-clock signals, the data may be overwritten normally.

Figure 3 shows the data-register cell for a TAP pin. An OR-gate permanently disables the output buffer if boundary-scan operation is selected. Consequently, it is impossible for the outputs in IOBs used by TAP inputs to conflict with TAP operation. TAP data is taken directly from the pin, and cannot be overwritten by injected boundary-scan data.

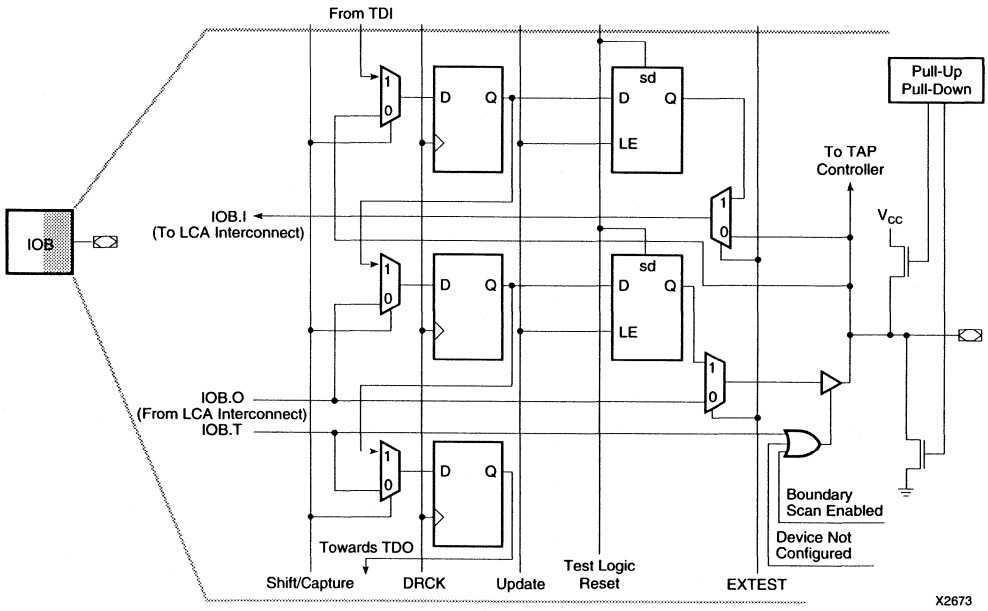


Figure 3. Boundary Scan Logic in a TAP Input IOB (TMS, TCK and TDI Only)

Table 2. Boundary Scan Order

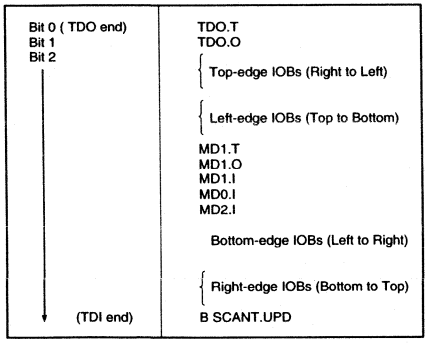


Table 2 lists, in data-stream order, the boundary-scan cells that make up the DR. The cell closest to TDO corresponds to the first bit of the data-stream, and is at the top of the table. This order is consistent with the BSDL description.

Each IOB corresponds to three bits in the DR. The 3-state control is first (closest to TDO), the output is next, and the input is last. Other signals correspond to individual register bits. IOB locations assume that the die is viewed from the top, as in XDE.

Note: All IOBs remain in the DR, independent whether they are actually used, or even bonded. Three bits, BSCANT.UPD, TDO.O and TDO.T, are included for Xilinx test purposes, and may be ignored by other users. CCLK, PROGRAM and DONE are not included in the boundary scan.

Tables in the data sheet show the DR order for all XC4000 family devices. The DR also includes the following non-pin bits: TDO.T and TDO.I, which are always bits 0 and 1 of the DR, respectively, and BSCANT.UPD which is always the last bit of the DR.

The Bypass Register

This is a 1-bit shift register that passes the serial data directly to TDO when a bypass instruction is executed.

User Registers

The XC4000 boundary-scan instruction set includes two USERCODE instructions, USER1 and USER2. Connections are provided to the TAP and TAP controller that, together with direct connections to the TAP pins, permit the user to include boundary-scan self-test features in the design.

The boundary scan block has six connections for user registers: SEL1, SEL2, TDO1, TDO2, DRCK and IDLE. TDI is available directly from the IOB that provides the TDI pin.

Note: The TDI signal supplied to user test logic is overwritten by boundary-scan test data during EXTEST. During user tests, it is not altered.

SEL1, SEL2 – SEL1 and SEL2 enable user logic. They are asserted (High) when the instruction register contains instructions USER1 and USER2, respectively.

TDO1, TDO2 – TDO1 and TDO2 are inputs to the TDO output multiplexer, permitting user access to the serial boundary-scan output. They are selected when executing the instructions USER1 and USER2, respectively. Input to user data registers can be derived directly from the TDI pin, thus completing the boundary-scan chain.

There is a one flip-flop delay between TDO1/TDO2 and the TDO output. This flip-flop is clocked on the falling edge of TCK.

DRCK – Data register clock (DRCK) is a gated and inverted version of TCK. It is provided to clock user test-data registers. TDI data should be sampled with the falling edge of DRCK (rising edge of TCK). The TDO output flip-flop accepts data on the rising edge of DRCK (falling edge of TCK). DRCK is active only during the Capture-DR and Shift-DR states of the TAP controller.

IDLE – IDLE is a second gated and inverted version of TCK. It is active during the Run-Test/Idle state of the TAP controller, and may be used to clock user test logic a set number of times, determined through TMS by the central test controller.

Using Boundary Scan

Boundary-Scan Availability

Full access to the built-in boundary-scan logic is always available between power-up and the start of configuration. Optionally, the built-in logic is fully available after configuration if boundary scan is specified in the design. At this time, user test logic is also available, and may be accessed through the boundary-scan port. During configuration, a reduced boundary-scan capability remains available: the SAMPLE/PRELOAD and BYPASS instructions only.

Figure 4 is a flow chart of the LCA start-up sequence that shows when the boundary-scan instructions are available. Since $\overline{\text{PROGRAM}}$ resets the TAP controller, boundary-scan operations cannot commence until $\overline{\text{PROGRAM}}$ has been taken High.

Full boundary-scan capabilities are then available until $\overline{\text{INIT}}$ is High. Without external intervention, $\overline{\text{INIT}}$ automatically goes High after ~1 ms. If more time is required for boundary-scan testing, $\overline{\text{INIT}}$ may be held Low beyond this period by applying an external Low signal to the $\overline{\text{INIT}}$ pin until testing is complete.

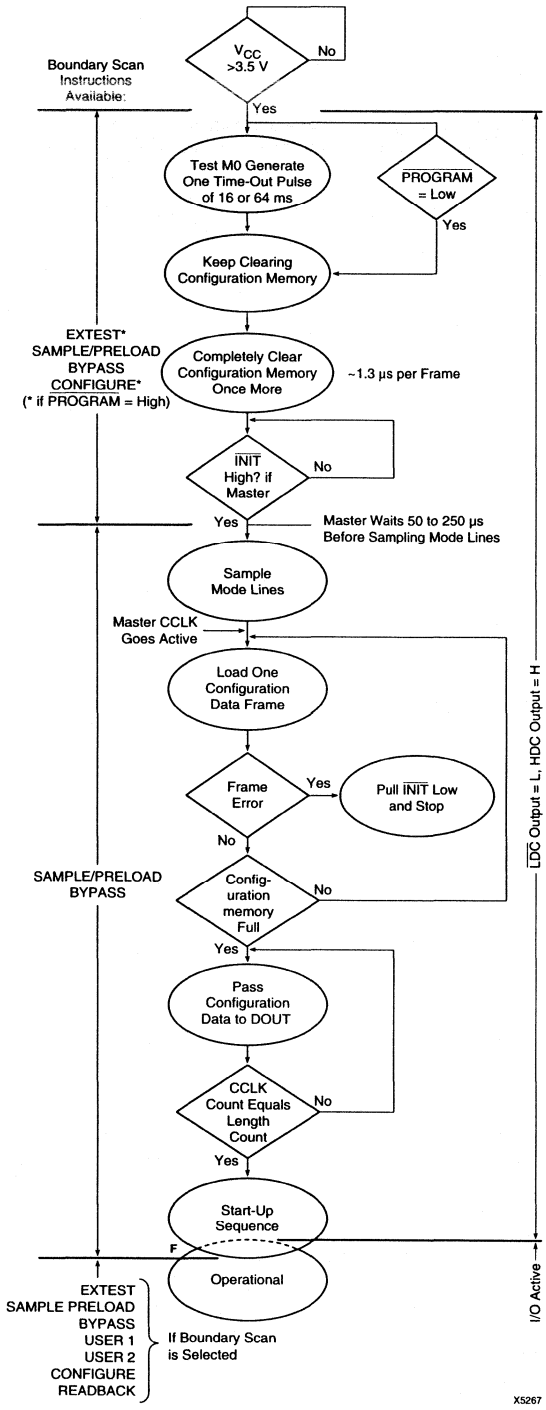


Figure 4. Start-up Sequence

The $\overline{INIT}$ pin is included in the boundary-scan coverage, and may be replaced by boundary-scan test data during an EXTEST instruction. *When performing an EXTEST prior to configuration, care must be exercised that boundary-scan input data does not force $\overline{INIT}$ High; this would terminate the EXTEST.*

During configuration, only SAMPLE/PRELOAD and BYPASS are available. Since the duration of the configuration period is determined by the configuration process itself, and this cannot be externally controlled, it is recommended that this period not be used for boundary-scan operations.

If boundary scan is not to be used after configuration, the start of configuration should be delayed until all boundary-scan operations are complete. This may be achieved by controlling $\overline{INIT}$, as described above. If boundary scan is enabled after configuration, unrestricted boundary-scan operations can be conducted once the configuration process is complete.

The exact point at which boundary-scan operations can be resumed after configuration (point F) depends upon the configuration mode. It is the point defined as Finished in the configuration timing diagram found in the Start-up section of the XC4000 Data Sheet.

The period of reduced boundary-scan availability is identified by a flag in the status word that is returned through the boundary-scan path whenever an instruction is loaded into the IR. The flag is High when all boundary-scan functions are available, and Low when only SAMPLE/PRELOAD and BYPASS are available. See the Instruction Register section.

Selecting Post-Configuration Boundary-Scan Operation

In a configured LCA device, the boundary-scan logic may or may not be active, depending on the configuration data loaded into the part. Activation of the boundary-scan logic, if desired, is part of the LCA design process. After configuration, boundary scan cannot be activated or de-activated without changing the configuration..

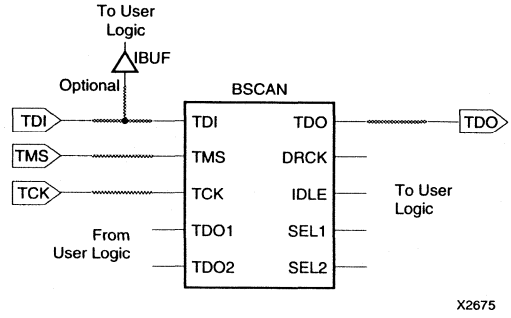


Figure 5. Boundary-Scan Schematic Symbols

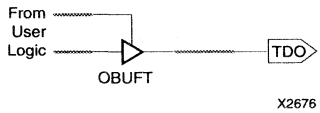


Figure 6. Typical Non-Boundary-Scan TDO Connection

If the BSCAN primitive is not included, boundary scan is not selected, and the IOBs used by the TAP inputs pins are freely available to PPR as general purpose IOBs. The TDO output pin may used as a logic output by explicitly connecting the TDO pad primitive to an OBUF or OBUFT as required, Figure 6.

Boundary scan may also be selected in the XACT Design Editor. The EditBlk command is used to change the configuration of the BSCAN block, found in the top left corner of the die. USED is toglgled so that it is highlighted. The TAP pins are permanently connected to the BSCAN block, although the connections are not explicitly shown. Connections to user test logic may be made using the design editor, if required.

XC4000 Boundary-Scan Instructions

The XC4000 boundary scan supports three IEEE-defined instructions, EXTEST, SAMPLE/PRELOAD and BYPASS, two user-definable instructions, USER1 and USER2, and two LCA-specific instructions, CONFIGURE and READBACK. The instruction codes are shown in Table 1.

EXTEST – While the EXTEST instruction is present in the IR, the data presented to the device output buffers is replaced by data previously loaded through the boundary-scan DR and stored in the update latch, Figure 7. Similarly, the output 3-state controls are replaced, and the data passed to internal system logic from input pins is replaced.

When a DR instruction cycle is executed, data arriving at the device input pins is loaded into the DR. The data from the system logic that drives output buffers and their 3-state controls is also loaded. This action occurs during the Capture-DR state of the TAP controller, Figure 1. Data is serially shifted out of the DR during the Shift-DR state; simultaneously, new data is shifted in. In the Update-DR state, the new data is transferred into the update latch for use as replacement data, as described above.

The replacement of system data with update latch data starts as soon as the EXTEST instruction is loaded into the IR. For this data to be valid, it must have been loaded by a previous EXTEST or SAMPLE/PRELOAD operation.

Since the DR and Update latch are modified during any DR instruction cycle, including BYPASS, the data in the update latch is only valid if it was loaded in the last DR instruction cycle executed before EXTEST is asserted.

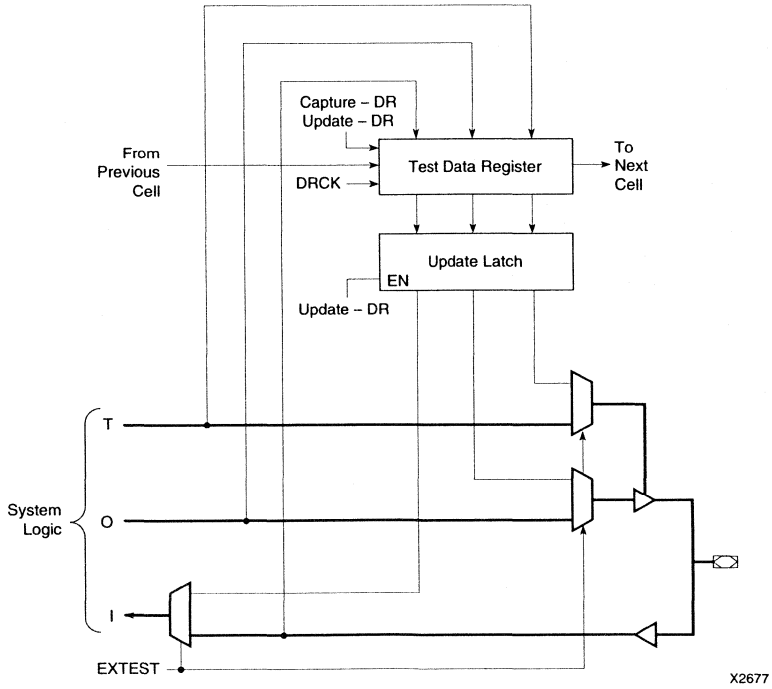


Figure 7. EXTEST Data Flow

The IEEE definition of EXTEST only requires that test data be driven onto outputs, that 3-state output controls be overridden, and that input data be captured. The capture of output data and 3-state controls and the forcing of test data into the system logic is normally performed during INTEST.

The XC4000 effectively performs EXTEST and INTEST simultaneously. This added functionality permits the testing of internal logic, and compensates for the absence of a separate INTEST instruction. However, when performing an EXTEST, care must be taken over what signals are driven into the system logic; data captured from internal system logic must be masked out of the test-data stream before performing check-sum analysis.

SAMPLE/PRELOAD – The SAMPLE/PRELOAD instruction permits visibility into system operation by capturing the state of the I/O. It also permits valid data to be loaded into the update register before commencing an EXTEST.

The DR and update latch operate exactly as in EXTEST, see above. However, data flows through the I/O unmodified.

BYPASS – The BYPASS instruction permits data to be passed synchronously to the next device in the boundary-scan path. There is a 1-bit shift register between the TDI and TDO flip-flop.

USER1, USER2 – These instructions permit test logic, designed by the user and implemented in CLBs, to be accessed through the TAP. Test clocks and paths to TDO are provided, together with two signals that indicate that user instructions have been loaded. For details, see the User Registers section above.

User tests depend upon CLBs and interconnect that must be configured to operate. Consequently, they may only be performed after configuration.

CONFIGURE – XC4000 LCA devices can be configured, or reconfigured through the TAP. Like EXTEST, this instruction is only available before $\overline{\text{INIT}}$ goes High or after a conventional configuration is finished.

After loading the CONFIGURE instruction, TCK clocks a normal configuration bit-stream into TDI while the TAP controller is in the Shift-DR state. The configuration preamble is passed to both TDO and DOUT. Configuration bits used by the device are not passed to the output, but are replaced by ones, as in a conventional configuration. Any bits beyond those required to configure the device are passed to TDO and DOUT.

READBACK – READBACK permits the configuration data of an LCA device to be read back through the TAP. This instruction differs from other boundary instructions in two ways.

- The readback logic is triggered (equivalent to Capture-DR) during the Update-IR state when the READBACK instruction is loaded. To re-trigger the readback logic, some other boundary-scan instruction must first be loaded, and then the READBACK instruction reloaded.
- TDI does not connect to the input end of the READBACK shift register. Consequently, data from upstream devices is lost.

For details of the readback bit-stream, see the Xilinx Application Note "Using the XC4000 Readback Capability" (XAPP 015).

Boundary Scan Description Language Files

Boundary Scan Description Language (BSDL) files describe boundary-scan-capable parts in a standard format used by automated test-generation software. The order and function of bits in the boundary-scan data register are included in this description.

BSDL files for XC4000 devices can be obtained from your Xilinx FAE, or by calling the Xilinx Applications Hotline. These files may also be downloaded from the Xilinx Technical Bulletin Board (BBS), and have filenames <device>.bsm.

Bibliography

The following publications contains information about the IEEE Standard 1149.1, and should be consulted for general boundary-scan information beyond the scope of this application note.

Colin M. Maunder & Rodham E. Tulloss. *The Test Access Port and Boundary Scan Architecture*. IEEE Computer Society Press, 10662 Los Vaqueros Circle, P.O. BOX 3014, Los Alamitos, CA 90720-1264.

John Fluke Mfg. Co. Inc. *The ABC of Boundary Scan Test*. John Fluke Mfg. Co. Inc., P.O. BOX 9090, Everett, WA 98206.

GenRad Inc. *Meeting the Challenge of Boundary Scan*. GenRad Inc., 300 Baker Ave., Concord, MA 01742-2174.

Ken Parker. *The Boundary Scan Hanbdbook*. Kluwer Academic Publications, (617) 871-6600.

Boundary-Scan Emulator for XC3000

XAPP 007.001

Application Note By BERNIE NEW

Summary

CLBs are used to emulate IEEE1149.1 Boundary Scan. The LCA device is configured to test the board interconnect, and then reconfigured for operation.

Specifications		*Xilinx Family*
Tests Supported	EXTEST	XC3000/XC3100
Number of CLBs	11 Core Logic 1/2 to 1-1/2 per IOB 1 per 3-State Control	*Demonstrates* State Machine Design

Introduction

With more complex integrated circuits and more densely packed PC boards, testability is a major issue. One solution to the testability problem is boundary scan. The XC4000-series LCA devices include boundary scan registers that meet the requirements of the IEEE1149.1 standard. While this standard provides for diagnostic testing and supports built-in self-test (BIST), one of its primary objectives is the testing of the interconnections between ICs. This is achieved using a mandatory external test mode, called EXTEST.

Although the XC3000-series LCA devices do not contain boundary-scan registers, it is possible to configure an XC3000 to emulate the EXTEST. This emulation consumes a significant amount of the LCA resources (almost all in an XC3020), and it is not suggested that boundary scan be built into a working design. However, because the RAM-based LCA device is reconfigurable, it can be configured for board testing, and then reconfigured for operation.

The second mandatory test mode, SAMPLE/PRELOAD, has no meaning because the LCA device must be reconfigured for testing. It is not, therefore, supported by the emulator. However, the minimum 2-bit Instruction Register provides four instructions to select between two choices, the Test Data and Output Registers. For consistency with other boundary-scanned parts, one of these instructions could be used to create a dummy SAMPLE/PRELOAD mode. Functionally, this would duplicate the EXTEST with the Test Data Register selected.

Four pins must be dedicated to the Test Access Port (TAP). Due to external interconnection requirements, these pins can probably not be reused in the actual design.

The TAP Controller, Instruction Register, Bypass Register and Test Data Output Buffer together with miscellaneous logic require 11 CLBs. The CLB requirement for the Test Data Register depends upon the number of IOBs used, and how they are configured. Each requires between 1/2 and 1-1/2 CLBs, plus 1 CLB for each distinct 3-state control. While this may not allow every IOB to be bidirectional with an independent 3-state control, it will accommodate most designs.

A specific boundary-scan emulation must be created for each LCA design. This comprises the 11 CLBs of core logic, which is common to all emulations, and a Test Data Register concatenated from four standard cells according to the output usage in the design. The output pins must be tied to match the design.

Operating Description

Overview

A block diagram of the IEEE1149.1 Boundary-Scan emulator is shown in Figure 1. The four pins used are Test Data In (TDI), Test Data Out (TDO), Test Mode Select (TMS) and Test Clock (TCK). Operation of the emulator is controlled by the TAP state machine. This, in turn, is controlled by the serial TMS data stream.

Test data is shifted from TDI, through either the Instruction or Test Data/Bypass Registers, to the TDO. The choice between Instruction and Test Data/Bypass Registers is made according to the TMS bit-stream. The Test Data or Bypass Register is selected by the contents of the Instruction Register.

Before shifting commences, input data is captured by a parallel transfer into the appropriate shift register. After shifting is complete, new data is transferred in parallel into a second register where it is available to the outputs.

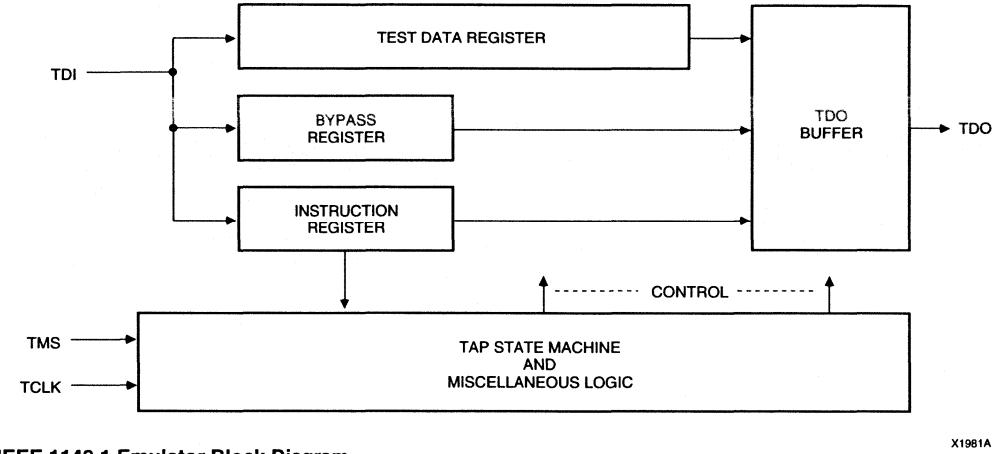


Figure 1. IEEE 1149.1 Emulator Block Diagram

After configuration, the emulator automatically enters the power-up state required by the specification, and therefore, the Test Reset Signal ($\overline{TRST}$) is not implemented. However, the polarity of all the registers is such that global reset may be used for this, if desired. The input pins used for TMS and TDI, and $\overline{TRST}$ if used, should be pulled up.

TAP Controller State Machine

The state diagram for the TAP Controller state machine is shown in Figure 2. This is implemented as two linked state machines, each using "one-hot" encoding.

The state-assignment table for this state machine is shown in Figure 3. Four state variables are used to create the states Test Logic Reset, Run Test/Idle, Select DR Scan and Select IR Scan.

In the latter two states, the second state machine may be initiated. This has six state variables, and creates the states Capture (CAP), Shift (SH), Exit1 (E1), Pause (PAU), Exit2 (E2) and Update (UPD). These are qualified by the output of the first state machine to control the Test Data and Instruction Registers as necessary.

While this second state machine in operating, the first state machine is held in its current state. Following the Update state, the first state machine is forced to the appropriate state determined by TMS.

Figure 4 shows the schematic diagram of the state machine, together with the equations that determine its next state. The only point of special interest is the use of clock enable in the first state machine. When the second state machine is in any of its first five states, the clock is disabled in the first state machine, thereby saving complexity in the next-state logic.

Note that the RTI flip-flop has inverters at its input and output. This causes the RTI state to be stored in active-Low form, such that this state is activated upon configuration or global reset. The pairs of flip-flops identified by circled numbers may be combined into single CLBs. The state machine requires six CLBs.

Instruction Register

The Instruction Register, shown in Figure 5, is two bits long, the minimum according to the specification. The shift register is enabled when the Instruction Register is selected by the state machine. In the Capture state, it is parallel loaded with 01(Binary), as required by the specification. It shifts data in the Shift state, and holds at other times.

Data from the shift register is clocked into the parallel register during the Update IR state. This parallel register is provided with a synchronous reset, which operates during the Test Logic/Reset state. The data in the parallel register is stored in inverted form, such that the Bypass Register (mandatory code: all ones) is selected after configuration or following a global reset.

For verification that the correct configuration has been loaded, additional bits could be added at the TDI end of the shift register. During Capture, these would be loaded with a code unique to the configuration. This would then be shifted out and become available as status bits. The parallel register need not be extended. Alternatively, the optional ID Code register could be implemented.

Test Data Register

The Test Data Register contains as many bits as there are used IOBs, plus one bit for each distinct 3-state control. This is concatenated from four types of 1-bit macros. Each of these is tied to a specific IOB, and the type of macro is determined by the function of the IOB.

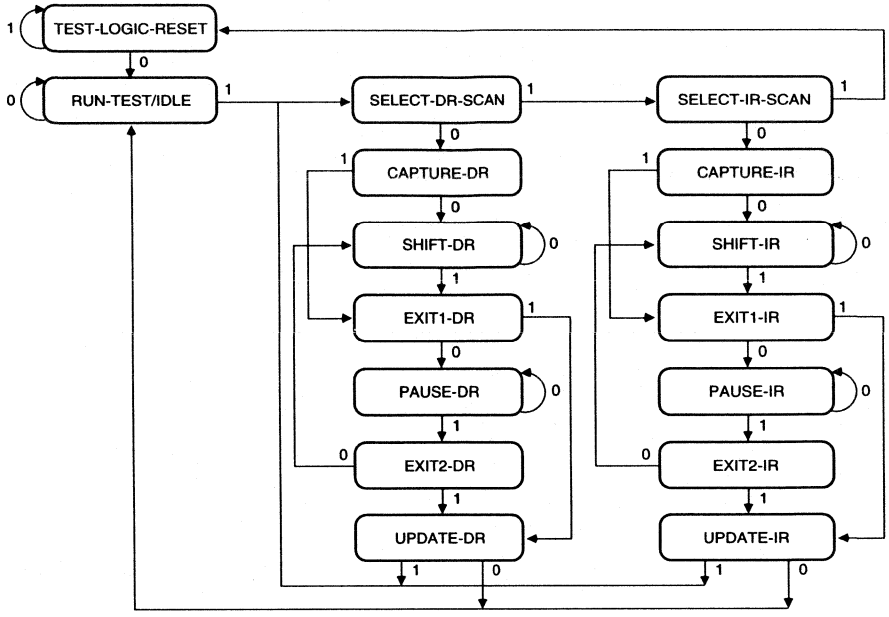


Figure 2. State Diagram for the TAP Controller

The simplest macro, shown in Figure 6, is used for input pins. Data from the pad is loaded during Capture, when the Test Data Register is selected. This macro uses 1/2 CLB.

Figure 7 shows the second macro, which also requires 1/2 CLB. Although this may be used for 3-state and bidirectional outputs, it is most appropriate for simple outputs. Data from the shift register is clocked into the IOB output flip-flop by Update DR. During Capture, data from the pad is loaded into the shift register.

If the output is enabled, as is always the case in a non-3-state pin, the data captured is the contents of the parallel register, provided it is not corrupted by an interfering external signal. If the 3-state output is not enabled, data is always captured from an external source; or it is undetermined if an external source does not exist.

A better output macro is shown in Figure 8. The IOB flip-flop is replaced with a CLB flip-flop. During Capture, the parallel register is always read back into the shift register. However, this macro requires 1 CLB per output.

This macro should also be used to control 3-state outputs. When the design gangs several outputs onto one 3-state control, only one of these macros need be used to control the ganged outputs.

The last macro, shown in Figure 9, is an enhanced macro for bidirectional pins This operates in the same way as the enhanced 3-state output macro, but has an additional multiplexer that selects between the input data and the

	TLR	RTI	DRS	IRS	CAP	SH	E1	PAU	E2	UPD
TEST LOGIC RESET	1	0	0	0	0	0	0	0	0	0
RUN TEST/IDLE	0	1	0	0	0	0	0	0	0	0
SELECT DR SCAN	0	0	1	0	0	0	0	0	0	0
CAPTURE DR	0	0	1	0	1	0	0	0	0	0
SHIFT DR	0	0	1	0	0	1	0	0	0	0
EXIT 1 DR	0	0	1	0	0	0	1	0	0	0
PAUSE DR	0	0	1	0	0	0	0	1	0	0
EXIT 2 DR	0	0	1	0	0	0	0	0	1	0
UPDATE DR	0	0	1	0	0	0	0	0	0	1
SELECT IR SCAN	0	0	0	1	0	0	0	0	0	0
CAPTURE IR	0	0	0	1	1	0	0	0	0	0
SHIFT IR	0	0	0	1	0	1	0	0	0	0
EXIT 1 IR	0	0	0	1	0	0	1	0	0	0
PAUSE IR	0	0	0	1	0	0	0	1	0	0
EXIT 2 IR	0	0	0	1	0	0	0	0	1	0
UPDATE IR	0	0	0	1	0	0	0	0	0	1

X1983

Figure 3. State Assignment for the TAP State Machine

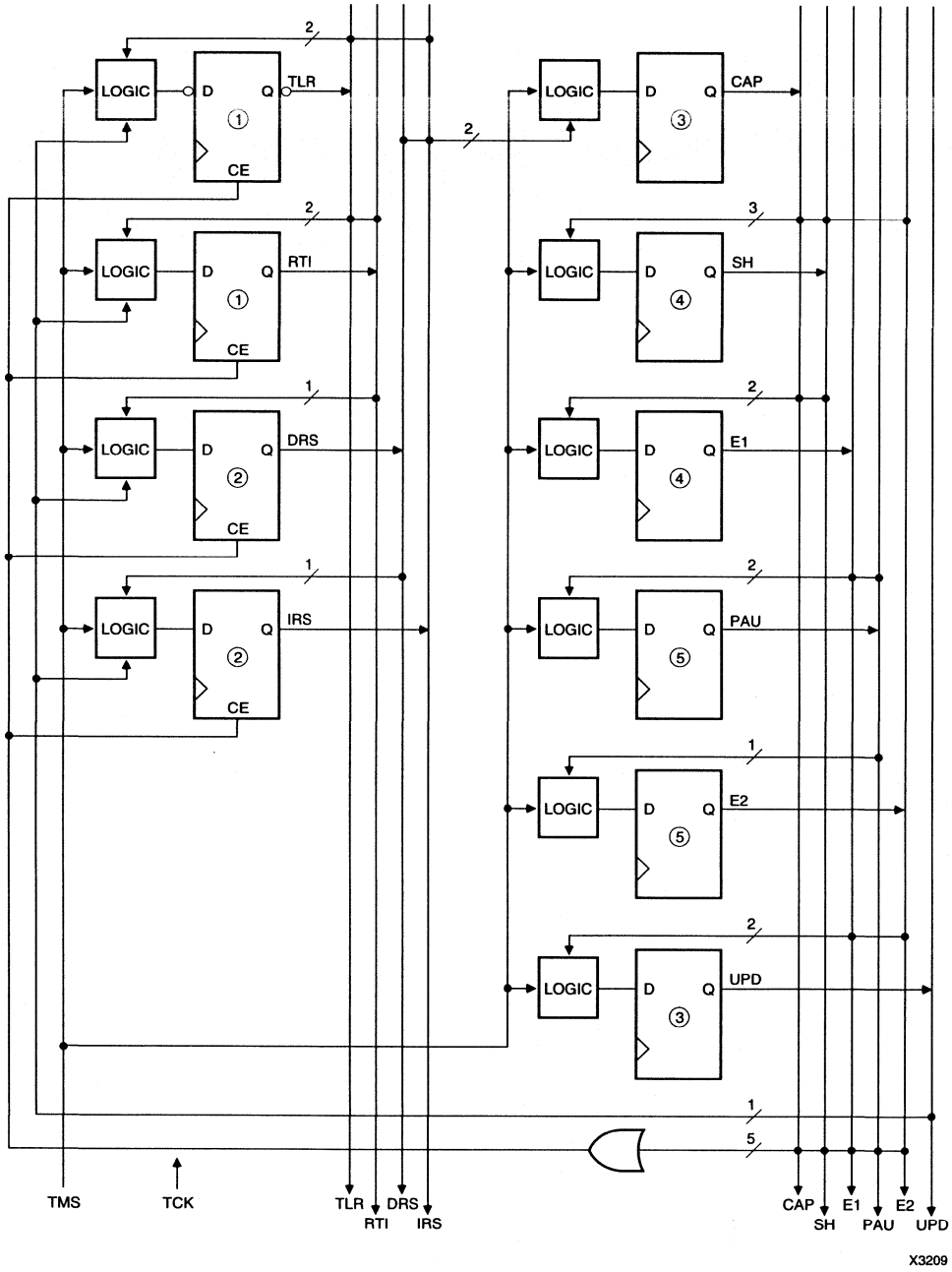


Figure 4a. TAP State Machine (6 CLBs)

$$TLR = CE[(IRS \bullet \overline{UPD} + TLR) \bullet TMS] + \overline{CE} \bullet TLR$$

$$RTI = CE[(TLR + RTI + UPD) \bullet \overline{TMS} + \overline{CE} \bullet RTI$$

$$DRS = CE \bullet [TMS \bullet (RTI + UPD) + \overline{TMS} \bullet DRS] + \overline{CE} \bullet DRS$$

$$IRS = CE \bullet [TMS \bullet (RTI + \overline{UPD}) + \overline{TMS} \bullet IRS] + \overline{CE} \bullet IRS$$

$$CAP = (DRS + IRS) \bullet \overline{TMS}$$

$$SH = (CAP + E2 + SH) \bullet TMS$$

$$E1 = (CAP + SH) \bullet TMS$$

$$PAU = (E1 + PAU) \bullet \overline{TMS}$$

$$E2 = PAU \bullet TMS$$

$$UPD = (E1 + E2) \bullet TMS$$

$$CE = CAP + SH + E1 + PAU + E2$$

X3409

Figure 4b. TAP State Machine Logic Equations

parallel register according to the 3-state control. This macro uses 1-1/2 CLBs.

Bypass Register

The Bypass Register, shown in Figure 10, operates when the Data Register is selected. A zero is loaded during Capture, and data is shifted through the register during Shift. Otherwise, the register holds. The Bypass Register uses 1/2 CLB

TDO Buffer

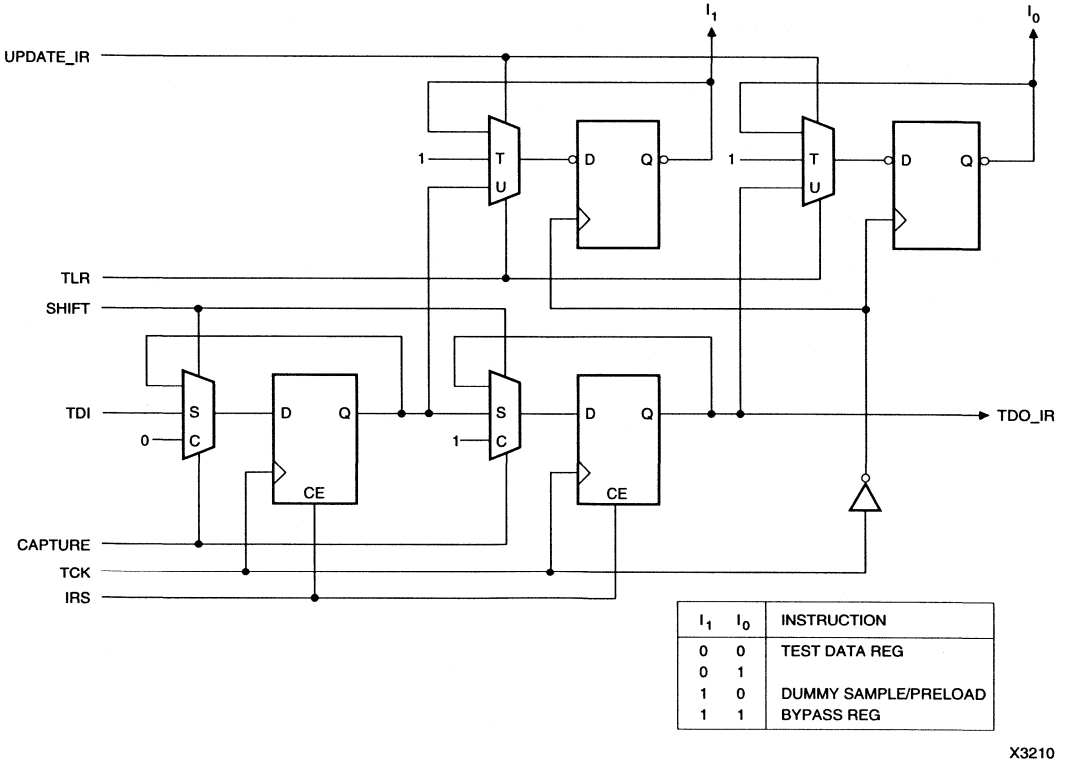
Figure 11 shows the Test Data Output Buffer. Data is selected from the Instruction Register, the Test Data Register or the Bypass Register, and clocked out on the negative edge of TCK. The 3-state output is only enabled during Shift. The TDO Buffer uses 1 CLB.

Miscellaneous Logic

The Miscellaneous Logic, shown in Figure 12., uses 1-1/2 CLBs. Its function is to combine states from the state machine to enable various registers.

Most registers in the emulator are clocked by TCK (or its inverse) and controlled by enables. The only exception is the IOB flip-flop used in the simple output macro of the Test Data Register. Since IOB flip-flops have no clock enable, a gated clock must used.

Rather than ANDing the clock with a gating signal, a flip-flop is used. During Update when the Data Register is selected, Update DR is clocked High on the negative edge of TCK. The state machine can only remain in the this state for one period, and this defines the length of the update clock. The ACLK buffer is used to distribute the Update DR clock.



I_1	I_0	INSTRUCTION
0	0	TEST DATA REG
0	1	
1	0	DUMMY SAMPLE/PRELOAD
1	1	BYPASS REG

X3210

Figure 5. Instruction Register (2 CLBs)

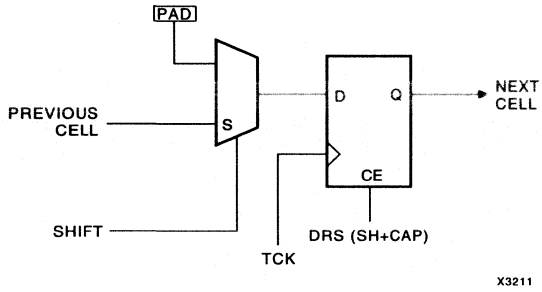


Figure 6. Data Register Input Cell (1/2 CLB)

Implementation Notes

The design support for the XC3000 Boundary-Scan Emulator comprises five soft macros. The first of these contains the 11 CLBs of core logic, including the Test Access Port. Location constraints must be added to the schematic to specify the desired location of the TAP input and output pins.

The remaining macros support different types of input/output pins. These macros need to be selected according to the input/output utilization, and connected to form a shift register between the data pins of the first macros. Again, a location constraint must be added to each macro, specifying the pin with which it is associated.

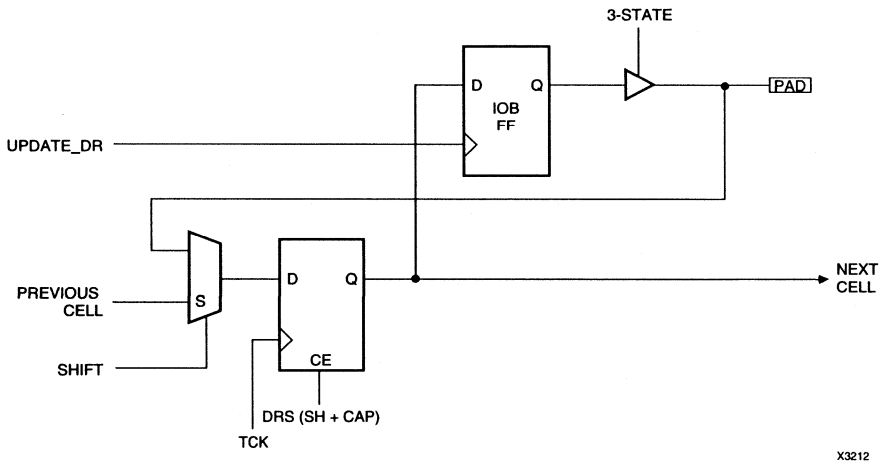


Figure 7. Data Register Output/Bidirectional Cell (1/2 CLB + IOB Flip-Flop)

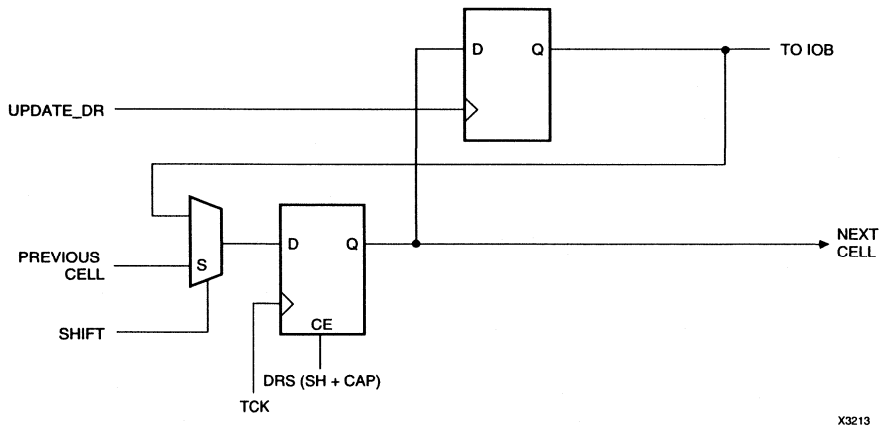


Figure 8. Data Register Enhanced Output/3-state Cell (2 bits/2 CLBs)

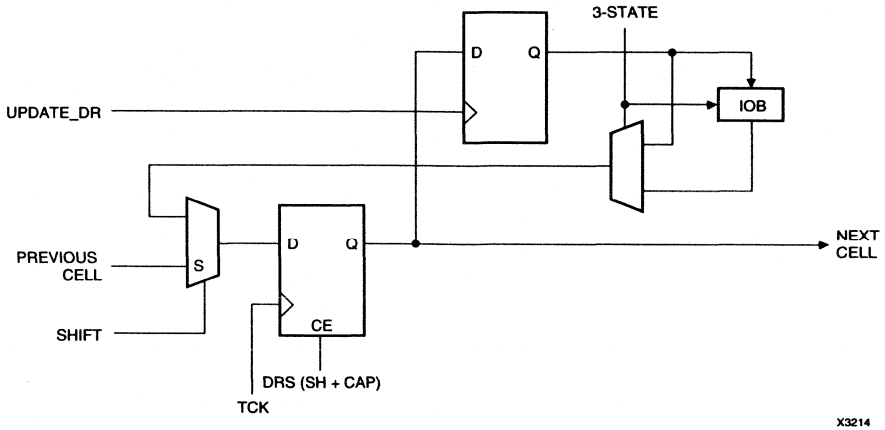


Figure 9. Data Register Enhanced Bidirectional Cell (1-1/2 CLBs)

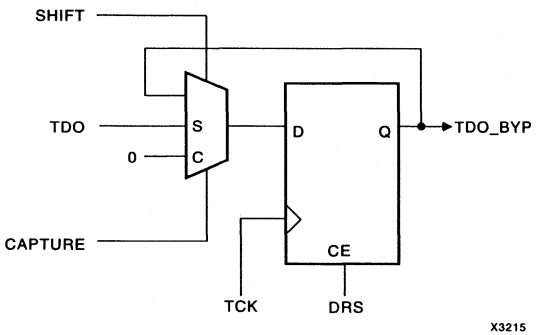


Figure 10. Bypass Register (1/2 BLB)

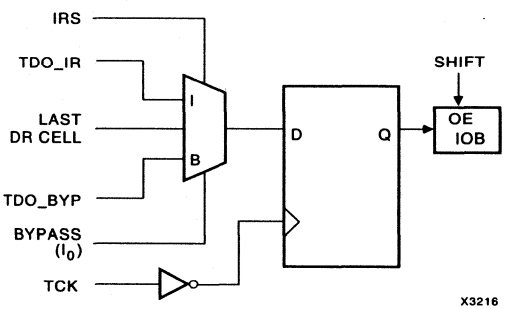


Figure 11. TDO Buffer (1 CLB)

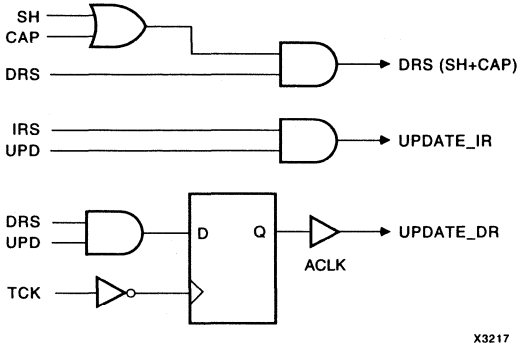


Figure 12. Miscellaneous Logic (1-1/2 CLBs)

Implementing Logic in the Universal Interconnect Matrix

XAPP 033.100 Application Note

Summary

This Application Note describes how the DS550 software implements logic functions using the AND capability of the Universal Interconnect Matrix.

Xilinx Family

XC7200/XC7300

Demonstrates

Universal Interconnect Matrix

Introduction

The Universal Interconnect Matrix (UIM) provides wire AND capability that can be used in various ways. This Application Note describes how to expand the input capacity of a Function Block, create an OR function using De Morgan's Theorem, and use the UIM as a decoder or signal blocker.

Using the UIM

Function-Block Input-Capacity Expander

Each function block has 21–24 inputs from the UIM. When the design equations contain common ANDed terms, these terms will be automatically factored out and ANDed in the UIM. This technique frees up Function Block inputs for use by other signals, as demonstrated by the counter design shown in Figure 1.

The carry signals can be explicitly expressed in PLUSASM™ with the following syntax

```
NODE (UIM) CARRY_8
           CARRY_16
EQUATIONS
CARRY_8    = Q0*Q1*Q2*Q3*Q4*Q5*Q6*Q7
CARRY_16   = CARRY_8*Q8*Q9*Q10*Q11*Q12*Q13
             *Q14*Q15
```

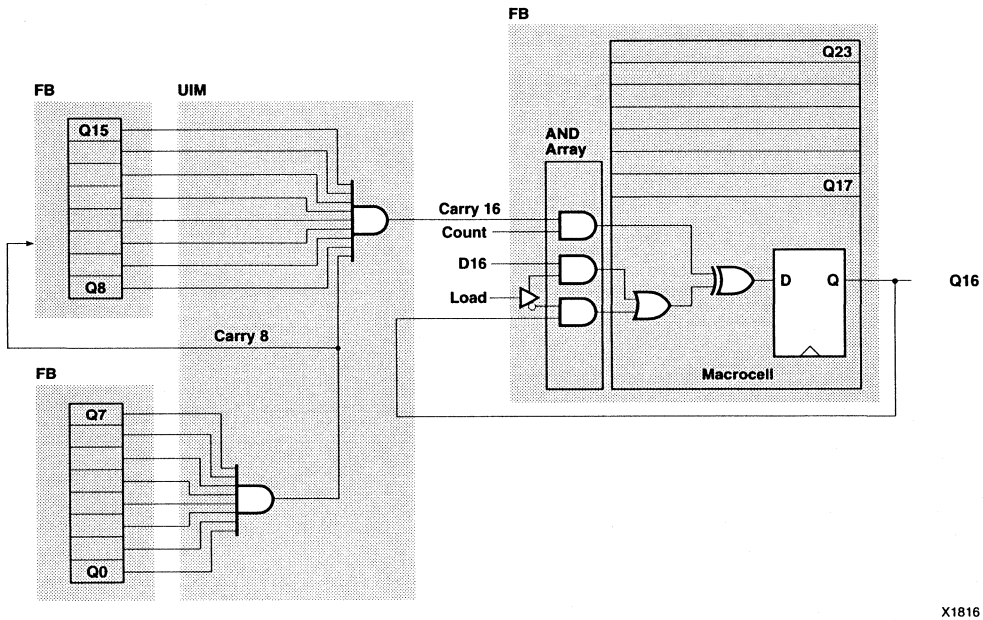


Figure 1. Function Block Input-Capacity Expansion

De Morgan OR Gates

An AND function in the UIM can be converted to an OR function using De Morgan's Theorem. Two or more Function Block outputs are inverted into the UIM and ANDed, as shown in Figure 2; the result is inverted at the Function Block inputs. Using De Morgan's Theorem, an AND with inverted inputs and outputs is equivalent to an OR.

In PLUSASM, this technique is explicitly implemented in two parts. First, a NOR function is created by inverting the inputs to create a UIM AND.

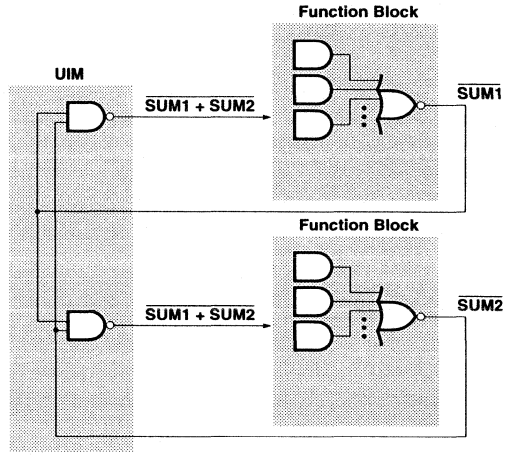
SUM1_NOR_SUM2 = $\overline{SUM1}$ * $\overline{SUM2}$

The inverse of SUM1_NOR_SUM2 is then used in subsequent equations.

D = $\overline{SUM1\_NOR\_SUM2}$ * ...

+ E * F ...

+



Functions larger than 16 P-terms split into intermediate sums joined by a negative-logic OR gate in UIM with no speed penalty.

X3212

Figure 2. Implementing a DeMorgan OR Gate

Decoding in the UIM

Figure 3 shows how to use the UIM in the XC7000 family as a decoder. The decoder output can feed directly into a Function Block without additional delay.

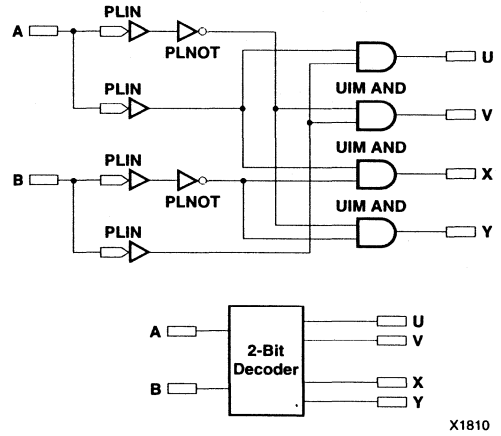
The PLUSASM equations for the decoder are as follows.

U = A * B

V = $\overline{A}$ * B

X = A * $\overline{B}$

Y = $\overline{A}$ * $\overline{B}$



X1810

Figure 3. UIM Decoder

Signal Blocker

A signal can be enabled or disabled in the UIM by ANDing it with a control signal.

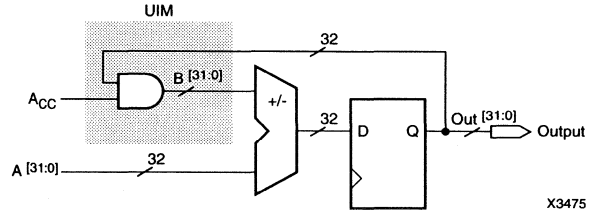
This technique is extremely useful when building Adder/Accumulators. The Accumulate function can be enabled with the control signal. A block diagram is shown in Figure 4.

This operation is expressed in PLUSASM by the following equations. NODE (UIM) B0 B1 B2...EQUATIONS

B0 = OUT0 * ACC

B1 = OUT1 * ACC

B2 = OUT2 * ACC



X3475

Figure 4. Implementing a 32-bit Registered Accumulator using the UIM

Comparison of XC3000 Counter Designs

XAPP 041.001

Application Note By BERNIE NEW

Summary

This Application Note discusses the functional, performance and density characteristics of the various counter designs available for the XC3000. Differences in these characteristics must be taken into account when choosing the most appropriate design.

Xilinx Family

XC3000/XC3100

Introduction

When selecting a counter design for a specific application, there are three primary considerations: does it meet the functional requirements, is it fast enough and could it use fewer LCA resources?

The functional requirements that must be considered include binary/non-binary operation, up, down and up/down counting, loadability, the provision of set/clear, count enable and synchronous operation to permit output decoding. Speed and resource utilization are self-explanatory, and can often be traded against each other. However, it must be realized that as a counter becomes more complex, it usually becomes both larger and slower.

Counter Designs

All the counters discussed in this Application Note have predictable binary-count sequences, and are fully synchronous designs. Table 1 summarizes the characteristics of the various counter designs. The same information is shown graphically in Figure 1.

Table 1. Counter Performance Summary

Counter Performance Summary

	Loadable	Up	Down	Up/ Down	8-Bit		10-Bit		12-Bit		16-Bit		20-Bit		24-Bit		32-Bit	
					MHz	CLBs	MHz	CLBs	MHz	CLBs	MHz	CLBs	MHz	CLBs	MHz	CLBs	MHz	CLBs
XC3100-3																		
XAPP 001		•			173	5	116	8	108	9	107	14	103	17	103	21		
XAPP 002	•	•	•	•	47	8	38	10	37	12	29	16	22	20	22	24		
XAAP 002	•	•	•	•							41	17						
XAPP 003	•	•	•		63	9			52	15	48	20						
XAPP 004		•	•								54	23					37	49
XAPP 004				•							46	27					37	56
XAPP 014		•									204*	24						
XC3000-125																		
XAPP 001		•			81	5	60	8	56	9	57	14	55	17	55	21		
XAPP 002	•	•	•	•	26	8	21	10	21	12	17	16	13	20	11	24		
XAPP 002	•	•	•	•							24	17						
XAPP 003	•	•	•		33	9			29	15	26	20						
XAPP 004		•	•								30	23					21	49
XAPP 004				•							25	27					20	56
XAPP 014		•									95*	24						

* Estimated

X3200

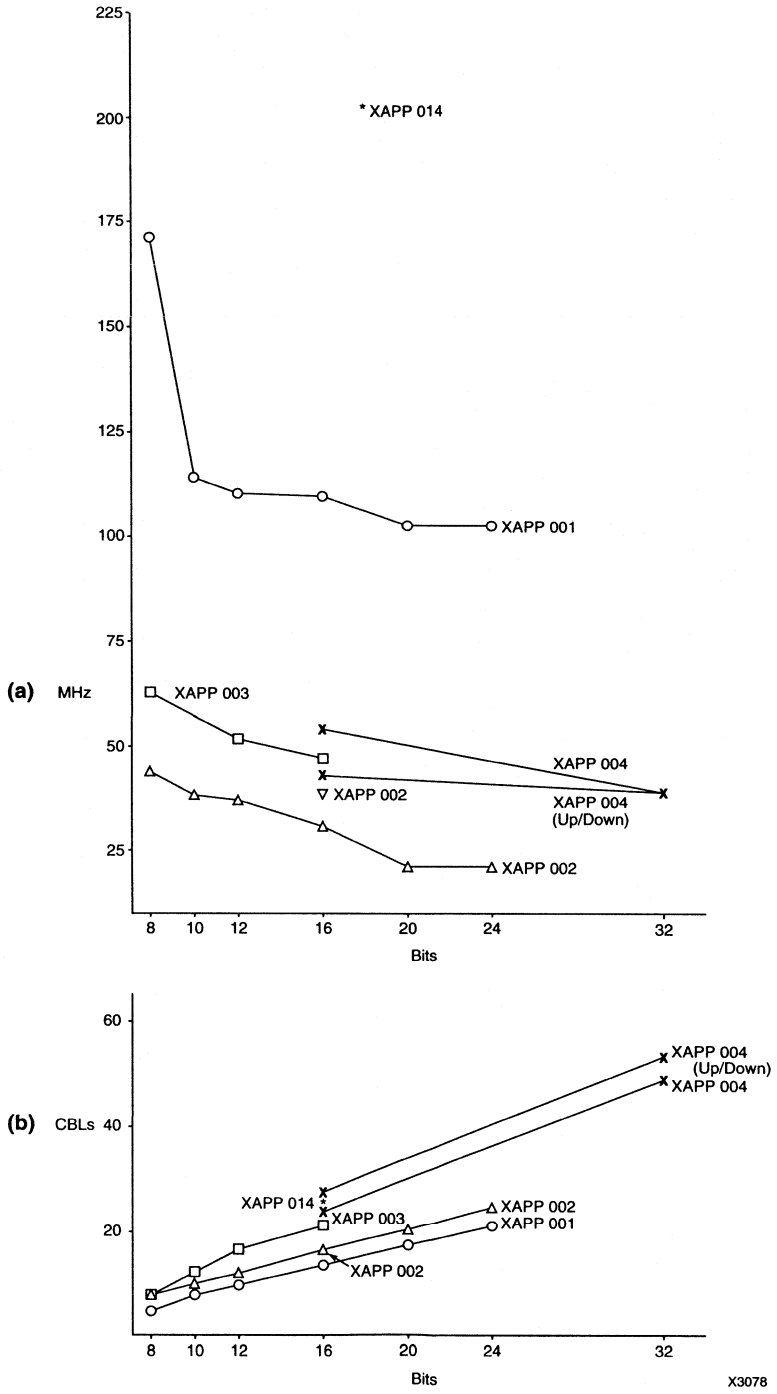


Figure 1. Counter Speed and Density (XC3100)

High-Speed Synchronous Prescaler Counter (XAPP 001)

This simple design provides a very basic non-loadable, up counter with a count-enable control. However, this simplicity permits it to be both the densest and the second fastest design.

A prescaler (CEP/CET) technique is used to gain speed, permitting the ripple-carry portion of the counter eight clock periods in which to settle. Without special adaptation, however, this technique precludes loading the counter. As a non-loadable counter, three bits can be implemented in three CLBs (1 CLB/bit), with the least significant six bits requiring only four CLBs; this explains the compactness. Only one T_{ILO} delay is incurred in the ripple-carry path for each three bits. This permits good speed to be maintained, even in long counters.

It is easy to convert the design into a down counter, but n not possible to convert it into an up/down counter.

Simple, Loadable, Up/Down Counter (XAPP 002)

Being loadable, this counter is unable to benefit from the prescaler technique, and a simple ripple-carry scheme is used throughout. Consequently, it is slower than the above design. The maximum clock frequency is inversely proportional to the length of the counter; the ripple-carry path incurs one T_{ILO} delay for each two bits.

With two CLBs required for each two bits, the CLB density is similar to the above counter (1CLB/bit). However, there is no equivalent reduction in complexity in the low-order bits, and the design, therefore, requires more CLBs.

The up/down-control logic is incorporated into the carry path, but does not impact the speed or the density; these attributes are determined by the number of outputs rather than the logic complexity. Optimal up counters and down counters can be implemented by simply tying the up/down control to the appropriate logic level. APR will eliminate any redundant logic, but the speed will not improve, nor will the CLB count decrease.

A modification to this counter almost doubles the maximum clock rate by dividing the carry path into two halves. The carry output of the lower half is used as a parallel count enable in the upper half. This use of a parallel count enable should not be confused with the prescaler technique; the carry path must still settle within one clock period. However, with this modification, it settles in approximately half the time. This technique effectively implements a conditional-sum incrementer within the counter.

This modification requires one additional CLB. Enable Clock is used for the parallel count enable, and the extra CLB is necessary to ensure that the clock is enabled during loading.

Synchronous Presettable Counter (XAPP 003)

In this design, speed is increased by replacing the serial gating of the ripple-carry path with parallel gating. Ideally, with arbitrarily wide gates, the carry-path settling time could be reduced to one gate delay.

However, with limited gate width, the settling time increases logarithmically with counter length; this is still a significant improvement over the linear increase seen previously, especially in longer counters. The additional speed is achieved at the cost of using more CLBs with more complex routing.

The specific implementation in the Application Note is for a modulo-N counter that could be used as a timer. The counter reloads whenever its terminal count is reached. To prevent loading from limiting the counters performance, detection of the terminal count is pipelined, permitting the load operation a full clock period.

The introduction of this pipeline stage essentially prevents the counter from being loaded at an arbitrary time. However, the pipeline could easily be removed for more general counter applications.

Loadable Binary Counter (XAPP 004)

The loadable binary counter also uses parallel gating to accelerate the carry path. In this case, however, a more structured approach is taken. A fast lookahead-carry technique is used, resulting in a carry path with a consistent depth of gating. Consequently, there are many equally critical paths.

The regular structure lends itself to hand placement when maximum speed is the objective. The irregularity and fewer critical paths of the previous design reduces its dependence on CLB placement. The previous design will, therefore, perform better using the automatic placement tools, and it is possible to improve its performance by rerouting a few critical paths. However, it will not match the performance of the current design when optimally placed.

Ultra-Fast Synchronous Counters (XAPP 014)

In some applications, such as clock division, the only requirement of a counter is that it count very fast. This counter is designed to fill that need. Compared to the first design described above, this design is approximately twice as fast, but uses almost twice as many CLBs.

The key to its high speed is the use of a prescaler technique, together with an "active Longline" to distribute the parallel count enable. This distribution scheme uses replicated flip-flops to eliminate the delay and depends, for its operation, upon the predictability of the binary sequence.

For a more detailed description of the above designs, see the individual Application Notes.

High-Speed Synchronous Prescaler Counter

XAPP 001.002

Application Note By PETER ALFKE AND BERNIE NEW

Summary

Borrowing the concept of Count-Enable Trickle/Count-Enable Parallel that was pioneered in the popular 74161 TTL-MSI counter, a fast non-loadable synchronous binary counter of arbitrary length can be implemented efficiently in XC3000-series LCA devices. For best partitioning into CLBs, the counter is segmented into a series of tri-bits. Design files are available for 8, 10, 12, 16, 20 and 24-bit versions of this counter.

Specifications			*Xilinx Family*
Length	8	16 Bits	XC3000/XC3100
Maximum Clock Frequency			*Demonstrates*
XC3100-3	173	107 MHz	
Number of CLBs	5	14	Fast Counter Technique

Introduction

Prescaler-counter designs originated with small, high-speed counters used to divide an incoming clock frequency and, thereby, provide a clock to a larger, slower counter. This scheme was adapted for use in cascading the synchronous 74161 counter.

The Terminal Count of the least significant 74161 was used as a parallel clock enable to the remaining counters. This effectively reduced the clock rate to those counters by a factor of 16, allowing their ripple-enable path 16 times longer to settle.

This only worked if the counter was not loaded. If it were, the first parallel enable would typically occur less than 16 clocks after the load. Depending on the value loaded, the ripple-enable path might not have time to settle.

Techniques exist to overcome this problem, but for a non-loadable counter they are unnecessary. This application note describes a 103-MHz 24-bit non-loadable counter, as shown in Figure 1. For optimal CLB usage, the counter is partitioned into 3-bit sections (tri-bits), the first of which acts as the prescaler.

Operating Description

The least significant tri-bit has a Count-Enable Output (CEO) that is routed to all the Count-Enable-Parallel (CEP) inputs in the rest of the counter.

The Count-Enable Output from any other tri-bit drives the next more significant Count-Enable-Trickle (CET) input. The clock causes any tri-bit to increment only if all its Count-Enable (CE) inputs are active. CEO is active when all three bits are set and CET is High. CEP does not affect CEO.

Using CEP, the least-significant tri-bit stops the remaining counter chain for seven out of eight clock pulses, allowing ample time for the CEO-CET ripple-carry chain to stabilize. The maximum clock rate is determined by the Clock-to-CEO delay of the first tri-bit ($T_{CKO} + T_{ILO}$), plus the CEP input set-up time of the other tri-bits (T_{ICK}) and the routing delay of the CEP net.

For a 24-bit counter in a -125 device, this critical delay can be less than 25 ns. The higher tri-bits are not speed critical if they propagate the CET signal in less than eight clock periods, easily achievable for counters as long as 20 tri-bits, i.e. 60 bits.

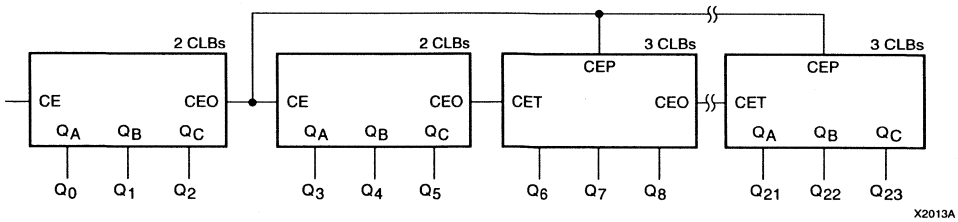


Figure 1. 50 MHz Non-Loadable Binary Counter

Supporting design files are available on the Xilinx Technical Bulletin Board under the name XAPP001V (Viewlogic)

As shown in Figure 2, the two least-significant tri-bits fit into two CLBs each. The higher tri-bits have two Count-Enable inputs (CEP and CET), and require three CLBs each.

For faster operation, it is possible to pipeline the CEP, separating into two clock periods the detection of Terminal Count in the first tri-bit and its distribution as CEP. The modification to the first tri-bit is shown in Figure 3. The state before Terminal Count is detected. The flip-flop is set for the duration of the terminal count state.

If this modification is used, the CET input to the first tri-bit will no longer act correctly as Count Enable for the entire counter. The EC pin on the CLBs should be used in its place.

Adding this pipeline stage reduces the critical CEP delay to 20 ns, and increases the maximum clock rate to 50 MHz. In an XC3100 the maximum clock rate is 103 MHz.

Implementation Notes

In this counter, the critical delay is the distribution of the CEP signal, and for maximum speed, this should use a Longline. Consequently, the counter should be partitioned using CLBMAPs and should occupy a row or column of CLBs.

Soft macros are available for 8, 10, 12, 16, 20 and 24-bit counters. A READ.ME file accompanying these macros describes the implementation.

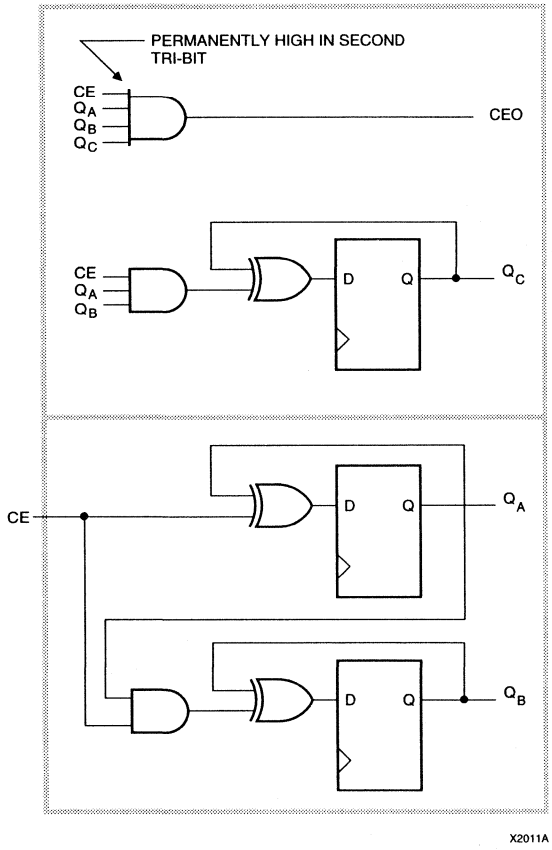


Figure 2a. First and Second Tri-bits Use Two CLBs Each

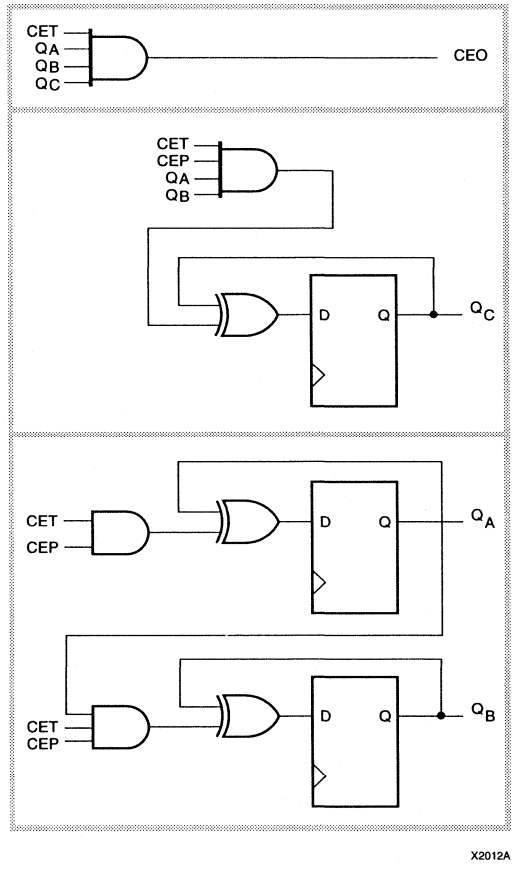


Figure 2b. All More Significant Tri-bits Use Three CLBs

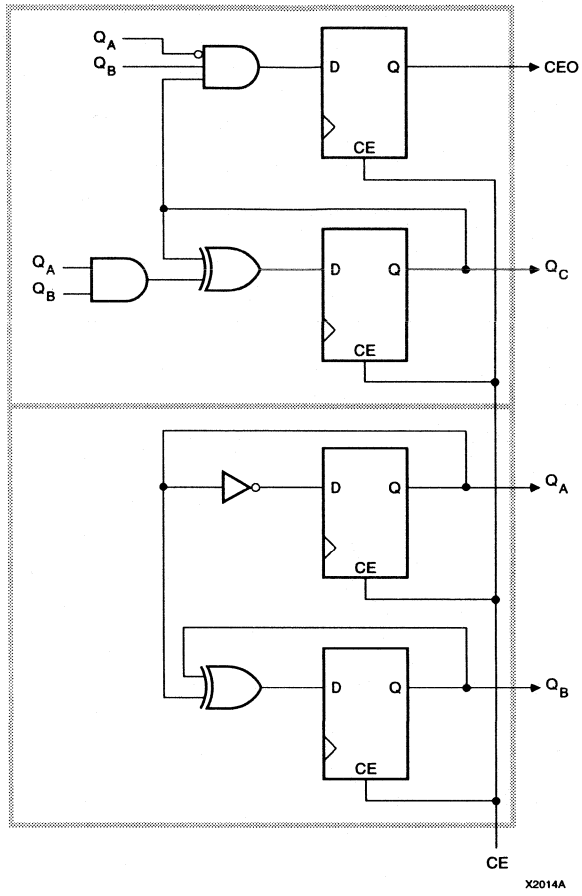


Figure 3. Least Significant Tri-bit with Pipeline

Simple Loadable Up/Down Counter

XAPP 002.002

Application Note By BERNIE NEW

Summary

The 5-input function generator of the XC3000 family CLB makes it possible to build fully synchronous, loadable up/down counters of arbitrary length. These use only one CLB per bit, and the ripple carry delay is only 1/2 T_{ILO} per bit. Design files are available for 8, 10, 12, 16, 20 and 24-bit versions of this counter. A 16-bit higher performance version is also available.

Specifications

Length	16 Bits
Maximum Clock Frequency XC3100-3	41 MHz
Number of CLBs	17

Xilinx Family

XC3000/XC3100

Demonstrates

Simple Counter Technique

Introduction

Using a ripple-carry technique, this simple counter requires the least number of CLBs of any XC3000 loadable counter. To improve its performance, the counter is partitioned into 2-bit segments. While the maximum clock frequency remains inversely proportional to the number of bits, this partitioning reduces the incremental delay to one T_{ILO} per bit-pair.

The performance of ripple-carry counters benefits greatly from hand-placement of the CLBs, in that zero-delay direct interconnects can be exploited in the critical carry path. With automatic placement, a more complex counter, with fewer routing delays in the critical path, offers better performance at the expense of using more CLBs.

If only an up counter or down counter is required, the up/down counter may be entered into the schematic with the up/down control tied to a logic one or zero as necessary. APR automatically eliminates the redundant logic to create the up or down counter. In this case, however, the number of CLBs cannot reduced, but routing resources are conserved. Up and down counter designs obtained in this manner cannot be improved upon if this ripple-carry technique is to be used.

The widely used CEP/CET prescaler technique for speed enhancement cannot be used in counters that are up/down or loadable. Up/down counters might reverse their direction of count at any time, and do not guarantee the ripple-carry chain a sufficient number of clock periods in which to stabilize. Similarly, loadable counters do not guarantee adequate time after a load.

However, some speed improvement may be gained by using CEP and CET. While this improvement is not as large as that offered by a prescaler, the clock rate is approximately doubled, and the additional resources consumed are minimal.

As shown in Figure 1, the counter is broken into equal halves. For up/down counting and loading, the ripple-carry path in both halves must settle within one clock period, as must the CEP distribution net. This distribution delay is in series with the carry path of the lower bits, and an unequal split might be used to compensate.

Operating Description

The basic counter cell, shown in Figure 2, uses two CLBs. The first CLB implements two T-type flip-flops. Along with the trigger inputs, these flip-flops have independent data inputs and a shared parallel enable to facilitate loading.

The second CLB implements two bits of the ripple-carry chain. The second carry bit is derived from two counter bits and a carry input. In this way, only one TILO delay is incurred per bit-pair. Down counting is achieved by inverting the counter bits into the carry chain. Counters

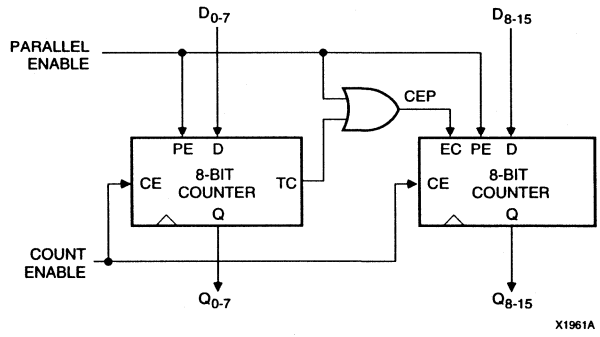


Figure 1. Loadable Counter Using CEP/CET

Supporting design files are available on the Xilinx Technical Bulletin Board under the name XAPP002V (Viewlogic)

of any length may be implemented by simply cascading as many of these 2-bit cells as necessary.

When implementing the CET/CEP version, the EC pin of the CLB is used for CEP. This necessitates an additional OR-gate to enable the clock during loading. If only an up-counter or down-counter is implemented, this OR-gate may be nested into the CLB that generates CEP.

Implementation Notes

To minimize the ripple-carry delay, the carry CLBs should be adjacent to each other, so that the carry can be propagated on zero-delay direct interconnects. Consequently, the counter should be partitioned using CLBMAPs and should occupy a row or column of CLBs. This organization also permits CEP to be distributed on a Longline, if needed.

Soft macros are available for 8, 10, 12, 16, 20 and 24-bit versions of the basic counter. A 16-bit counter using the CET/CEP technique is also available as a soft macro. A READ.ME file accompanying these macros describes the implementation.

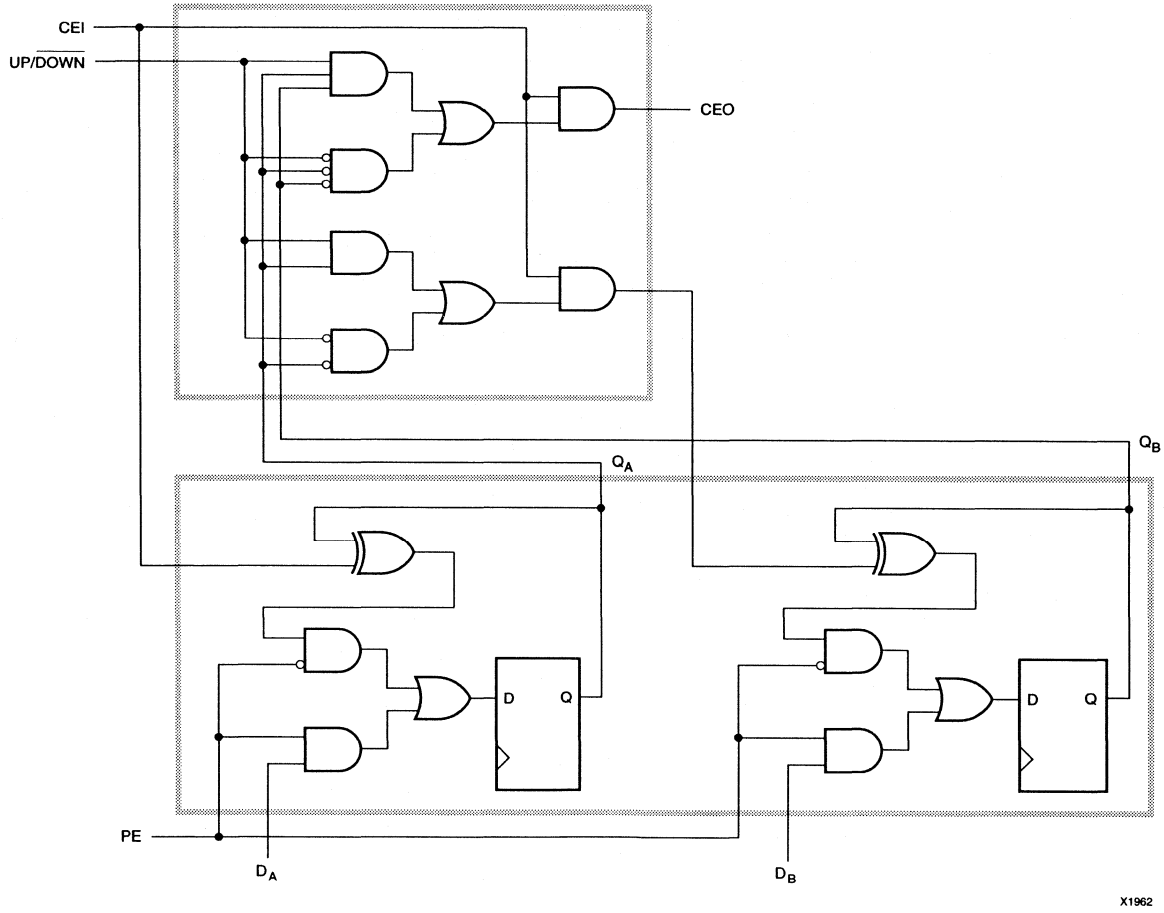


Figure 2. 2-Bit Counter Cell

Synchronous Presettable Counter

XAPP 003.002 🖫

Application Note By PETER ALFKE AND BERNIE NEW

Summary

Presettable synchronous counters are implemented, where the carry path utilizes parallel gating to replace the serial gating found in ripple-carry counters. The result is fewer CLB delays in the critical path, but more CLBs are used and the routing is less regular. Design files are available for 8, 10, 12, 16, 20 and 24-bit versions of this counter.

Specifications			*Xilinx Family*
Length	8	16 Bits	XC3000/XC3100
Maximum Clock Frequency XC3100-3	63	48 MHz	*Demonstrates*
Number of CLBs	9	20	Fast Counter Technique

Introduction

In most counters, the maximum operating frequency is determined by the time it takes the carry path to settle after a clock. Given the new state, each bit must decide whether or not to toggle on the next clock. If the information, on which this decision is based, does not reach the bit in time, the counter will malfunction.

In ripple-carry counters, this information is passed from bit to bit through a chain of AND gates. While this structure can be exploited to obtain very fast routing, the delay still becomes prohibitive in longer counters.

The counter described in this Application Note replaces the chain of AND gates with an AND-gate tree. Data must pass through fewer gates to reach its destination and the carry path settles faster.

The irregular structure of the counter makes it difficult to establish an optimum placement. However, the fewer routing delays in its critical path reduce the dependence on good placement. This makes it ideal for use with automated design tools.

The counter detects Terminal Count and loads the value applied to its parallel input. This allows the counter to operate with any modulus. Two versions of the counter are described: an up counter and a down counter.

Operating Description

An 8-bit version of the counter is shown in Figure 1. The basic counter cell is two loadable T-type flip-flops implemented in a single CLB. The trigger inputs are driven from the array of AND gates, that combine all lower bits into a trigger input. In the 8-bit case, there are no more than two levels of AND gates.

A point of interest is the pipelined Terminal Count. Instead of detecting all ones, a value is detected one count earlier, and a flip-flop is set during the all ones state. Normally, TC must settle and be distributed as Parallel Enable within one clock period. The pipeline separates these two functions, and increases the maximum clock rate.

There is one trivial disadvantage to using this pipeline: if the counter is loaded with all ones, it does not load again on the following clock. Instead, it rolls over to all zeros and counts until it again reaches terminal count.

The TC pipeline flip-flop is provided with a Reset. This is intended for use immediately after power-up. It eliminates the potentially long delay before the first TC. Until the first TC, the counter cycle is not controlled by the load value.

Figure 2 shows how the counter may be converted to a down counter. The only change is to invert all the inputs to the AND gates, including the 1-input AND gate that drives the trigger input of bit 1. This inversion is absorbed into the counter cell.

Implementation Notes

For optimum partitioning, this counter should be implemented using CLBMAPs. Soft macros are available for 8, 12 and 16-bit up counters and down counters. A READ.ME file accompanying these macros describes the implementation.

Enhancements

The counter may be modified such that it may be loaded at any time, not just at Terminal Count. The Load command is used to reset the TC pipeline flip-flop. This causes the active-Low Parallel Enable to be asserted and the counter loads on the next clock pulse.

However, this technique must be used cautiously. If the flip-flop is reset when the count is one before TC, the Parallel Enable is asserted for two clocks. If this situation cannot be avoided, the active-Low Parallel Enable must be ANDed with Terminal Count at the input to the flip-flop, thus ensuring that a second load cannot occur.

🖫 Supporting design files are available on the Xilinx

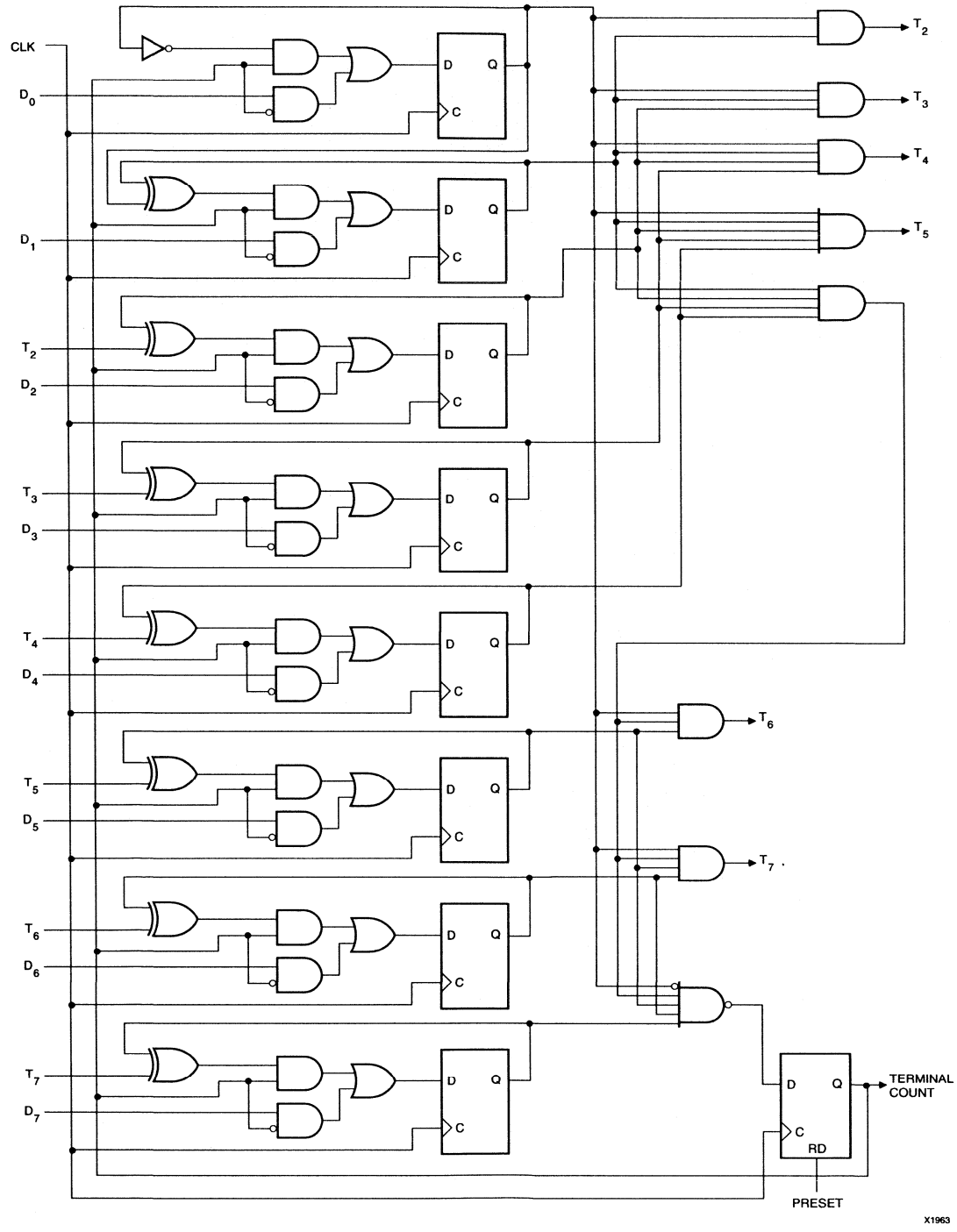


Figure 1. Presettable Up Counter

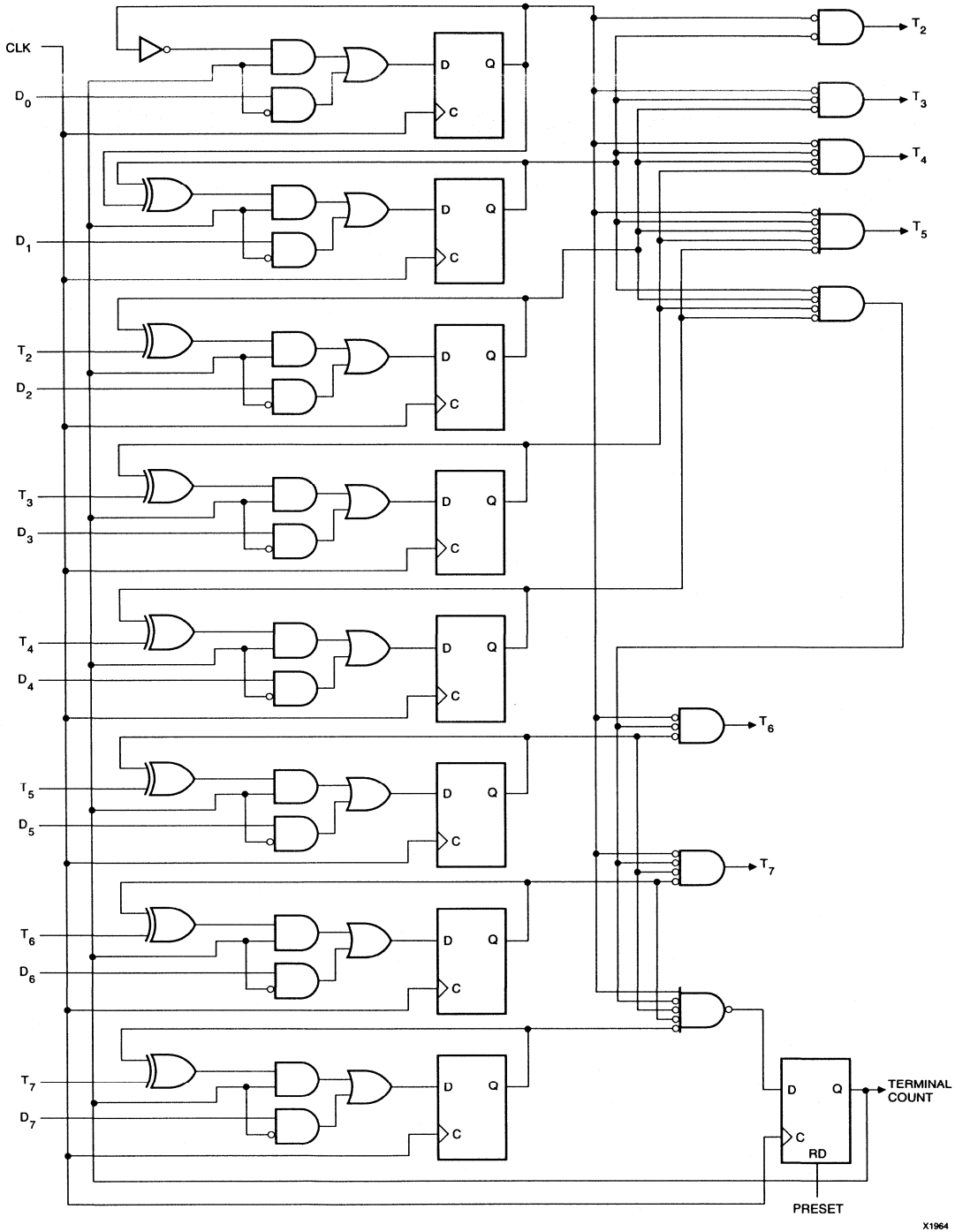


Figure 2. Presettable Down Counter

Loadable Binary Counters

XAPP 004.002 Application Note By BERNIE NEW

Summary

The design strategies for loadable and non-loadable binary counters are significantly different. This application note discusses the differences, and describes the design of a loadable binary counter. Up, down and up/down counters are described, with lengths of 16 and 32 bits. Design files are available for all six versions.

Specifications

Length	16 Bits
Maximum Clock Frequency XC3100-3	54 MHz
Number of CLBs	23

Xilinx Family

XC3000/XC3100

Demonstrates

Fast Counter Technique

Background

When designing a non-loadable counter, the fastest designs use some form of prescaler technique to exploit the fact that the more significant bits toggle much more slowly than the less significant bits.

The carry chain for the first few bits of the counter can usually be implemented in parallel and is very fast. However, the carry chain for the more significant bits usually requires multiple levels of gating and is much slower. Using prescaler techniques, the counter can operate at the speed of the less significant bits, by giving the more significant bits several clock periods in which to settle.

Typically, a 2- or 3-bit prescaler generates a high-speed count-enable signal that is broadcast through the more significant bits every four or eight clocks. In between these enables, the more significant bits are stable; the carry chain for these bits, therefore, has four or eight clocks periods in which to settle, instead of one.

These techniques depend upon the predictability of the binary sequence, and the implied low-speed operation of the more significant bits. When a counter is loaded, however, the binary sequence is disturbed, and its predictability is lost. To ensure correct operation following a load, the carry chain for the entire counter must settle before the next clock.

This reduces the speed of a prescaler counter significantly. Its operating frequency becomes constrained by the slow more significant bits rather than by the fast prescaler.

There are techniques such as pulse-swallowing and state-skipping that can be used to load a prescaler counter without loss of speed. However, these result in non-binary operation for a short time after loading, and some load values are not permitted.

Loadable Binary Up Counter

When designing a loadable binary counter, emphasis must be placed on balancing the carry delays. Unlike the prescaler counter, high-speed paths are of no benefit, and slow paths cannot be hidden. Figure 1 shows a good example of a loadable binary counter.

This counter is based on a 2-bit cell, as shown in Figure 2. The two bits are implemented in two CLBs, using loadable T-type flip-flops. Only one carry-in is required, the second carry-in being derived within the cell. The CLB clock enable may be used as Count Enable; however, the bits cannot be loaded while disabled. To overcome this, Parallel Enable must be ORed into the Count Enable line.

To form the carry chain, output bits are ANDed into groups of two and four, using the propagate cell shown in Figure 3. The propagate outputs are then ANDed together to form the even carries, according to the formulae of Table 1. Carries to the odd-weighted bits are generated within the counter cell.

With the exception of the trivial less significant bits, all carry delays comprise two levels of combinatorial CLB. This is longer than the direct paths from the less significant bits found in prescaler counters. However, prescaler counters typically have longer more-significant-bit delays, which is the chief speed constraint of a loadable counter.

The partitioning of the carry logic into the CLBs allows the counter to be implemented in an N-shaped configuration. A suggested placement of the CLBs is shown in Figure 4. Restricting the carry chain to a 2 x 4 block of CLBs minimizes the routing delays among them. With this organization, simulations show the counter will operate at 54 MHz.

Supporting design files are available on the Xilinx Technical Bulletin Board under the name XAPP004V (Viewlogic)

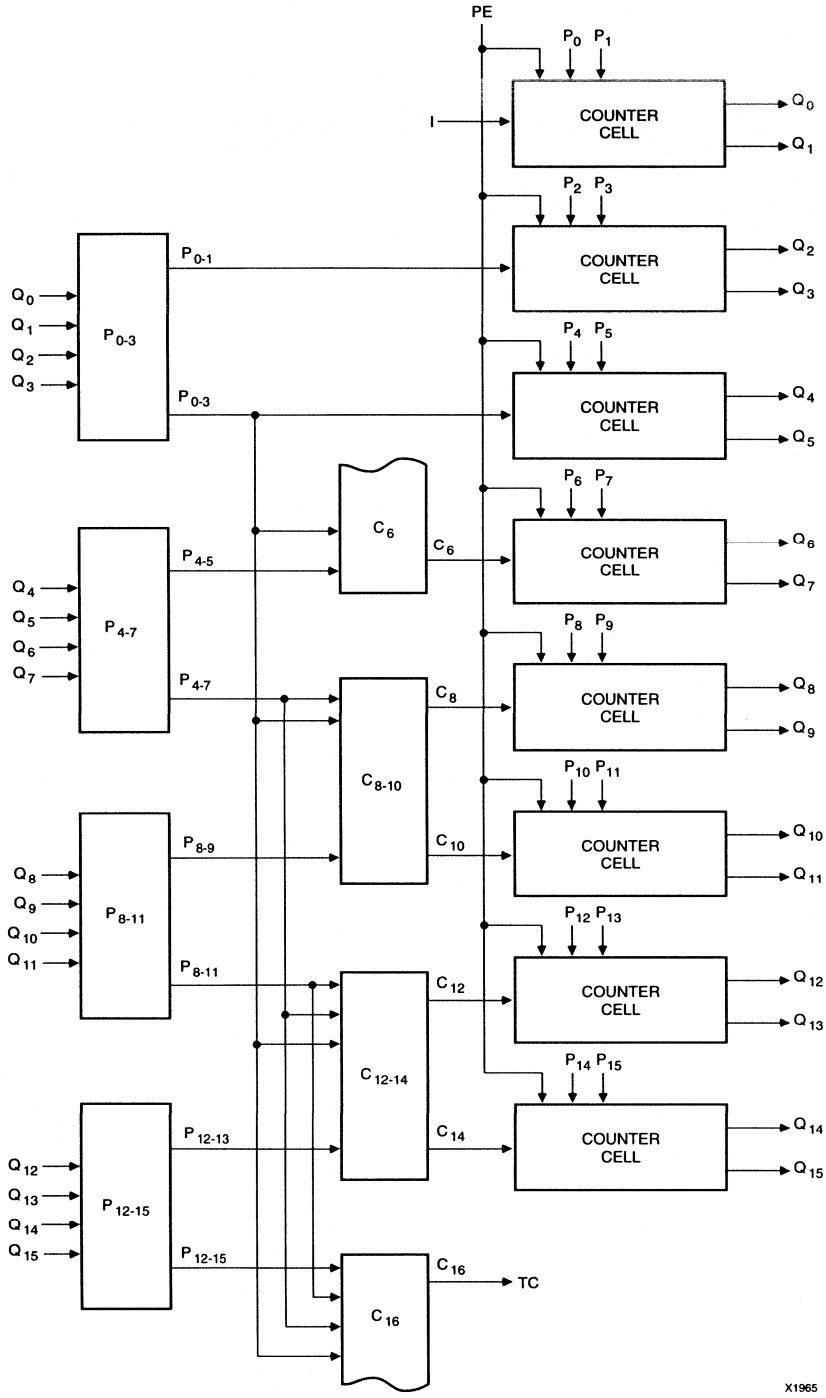


Figure 1. 16-Bit Loadable Binary Counter

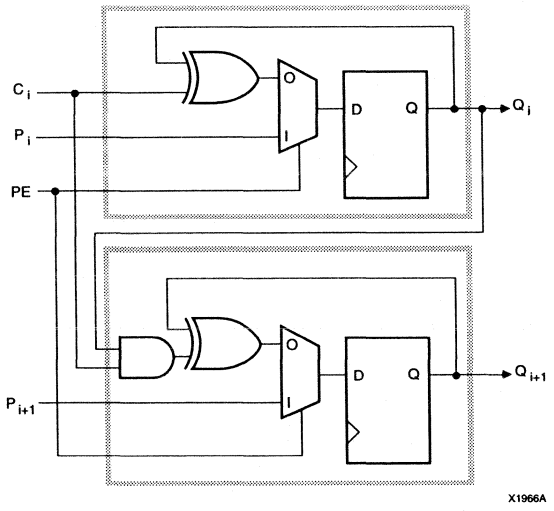


Figure 2. 2-Bit Counter Cell

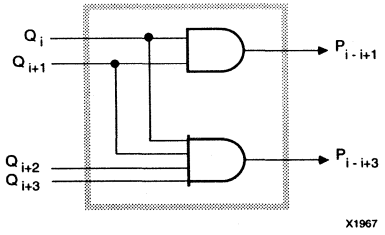


Figure 3. Propagate Cell

$$C_6 = P_{0\text{-}3} \bullet P_{4\text{-}5}$$
$$C_8 = P_{0\text{-}3} \bullet P_{4\text{-}7}$$
$$C_{10} = P_{0\text{-}3} \bullet P_{4\text{-}7} \bullet P_{8\text{-}9}$$
$$C_{12} = P_{0\text{-}3} \bullet P_{4\text{-}7} \bullet P_{8\text{-}11}$$
$$C_{14} = P_{0\text{-}3} \bullet P_{4\text{-}7} \bullet P_{8\text{-}11} \bullet P_{12\text{-}13}$$
$$C_{16} = P_{0\text{-}3} \bullet P_{4\text{-}7} \bullet P_{8\text{-}11} \bullet P_{12\text{-}15}$$

X2652

Table 1. Carry Logic Equations

Q_0	Q_1		$C_{8/10}$	Q_8	Q_9
Q_2	Q_3	$P_{0\text{-}3}$	$P_{8\text{-}11}$	Q_{10}	Q_{11}
Q_4	Q_5	$P_{4\text{-}7}$	$P_{13\text{-}15}$	Q_{12}	Q_{13}
Q_6	Q_7	$C_{6/16}$	$C_{12/14}$	Q_{14}	Q_{15}

X1969A

Figure 4. CLB Placement

For an 18-bit counter, C_{16} may be used as carry-in to bits 16 and 17, as shown in Figure 5. Additional TC logic must also be included. This extension does not involve additional levels of logic, but may incur additional routing delays.

The 18-bit counter may easily be extended to 32 bits by replicating bits 4 through 17, and using TC/C_{18} in the upper section in place of what was $P_{0\text{-}3}$. This entails one additional combinatorial delay, which reduces the maximum operating frequency to 37 MHz.

If this additional delay is unacceptable, two 16-bit counters may be concatenated, using C_{16} as the clock enable to the counter bits in the upper half. However, this creates two problems. Clock enable can no longer be used to provide count enable, and the counter may only be loaded when the lower half is at terminal count.

Both of these problems can be overcome separately, but not together. If C_{16} is moved to a separate CLB, a fifth input may be added. This could be Count Enable, which should be ANDed with the existing C_{16}, or Parallel Enable which should be ORed with it.

Loadable Binary Down Counter

If the counter bits are viewed as T-type flip-flops, the purpose of the carry chain is to determine which bits of the counter are to be toggled. For an up counter, a contiguous group of bits is toggled, starting with the least significant bit and extending up to, and including, the first zero. For a down counter, this group extends up to, and includes, the first one. The operation of the carry chain is the same in each case, but with the role of input ones and zeros reversed. Consequently, an up-counter may be converted into a down counter by simply inverting the output bits into the carry chain.

This requires two modifications to the up counter. First, all inputs to the propagate cells must be inverted, as shown in Figure 6. Second, the counter cell must be modified so that the direct path from the even bit to the odd bit becomes inverting, as shown in Figure 7. In all other respects, the counter remains the same. Performance and expandability are unaffected.

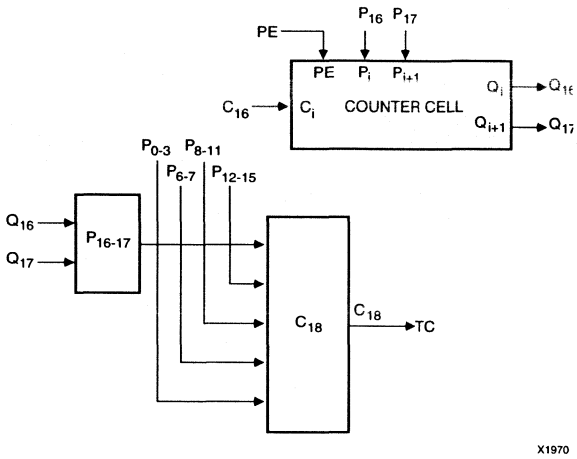


Figure 5. Extension to 18 Bits

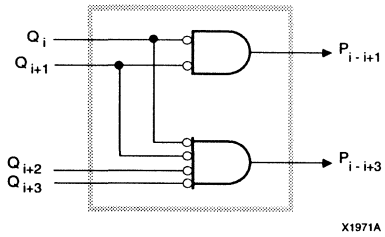


Figure 6. Down-counter Propagate Cell

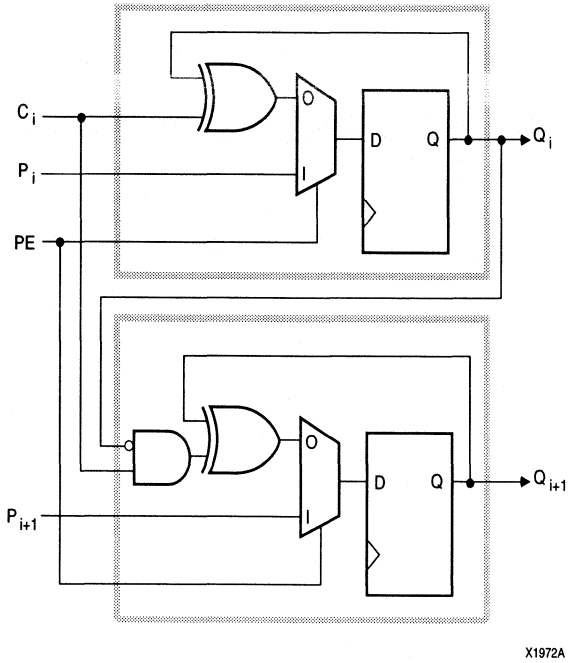


Figure 7. 2-Bit Down-counter Cell

Loadable Binary Up/Down Counter

To create an up/down counter, simply make the above inversions programmable. For the counter bits, this is not a problem. An XOR gate is placed in the direct path, as shown in Figure 8.

The propagate cells are more of a problem. The 2- and 4-input functions become 3- and 5-input when the up/down control is added; they can no longer share a single CLB.

The propagate cells must be split in two CLBs each, and the 3-input functions combined if necessary. Two or four additional CLBs are required, and additional routing delays might be created due to the higher fan-outs and the longer signal paths among the greater number of CLBs.

This design results in 16-bit up/down counters that operate at 46 MHz, and 32-bit up/down counters that operate at more than 37 MHz

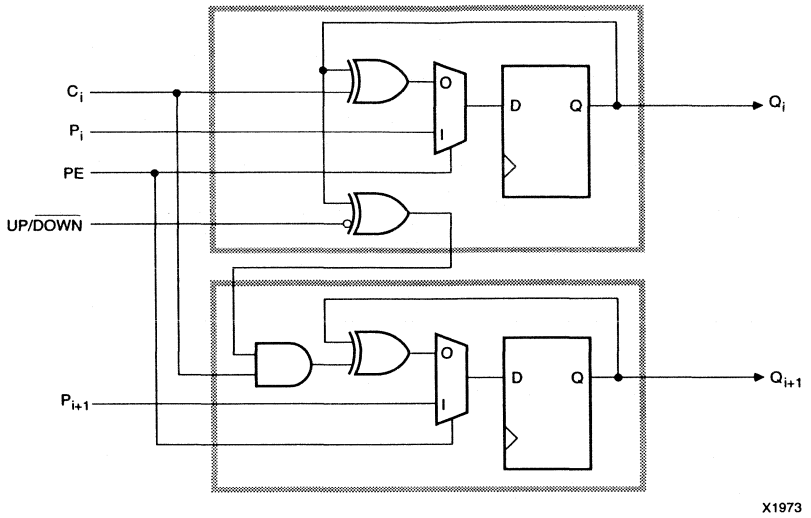


Figure 8. 2-Bit Up/Down-counter Cell

Ultra-Fast Synchronous Counters

XAPP 014.001

Application Note By BERNIE NEW

Summary

This fully synchronous, non-loadable, binary counter uses a traditional prescaler technique to achieve high performance. Typically, the speed of a synchronous prescaler counter is limited by the delay incurred distributing the parallel Count Enable. This design minimizes that delay by replicating the LSB of the counter. In this way even the small Longline delay is eliminated, resulting in the fastest possible synchronous counter.

Specifications

	XC4000 (-5)	XC3100 (-3)	XC3000 (-125)	
Counter Length	16	16	16	Bits
Maximum Clock Frequency	111	204	95	MHz
Number of CLBs	17	24	24	

Xilinx Family

XC3000/XC3100
XC4000

Demonstrates

Ultra-fast Counter Design

Introduction

The use of a prescaler is a common technique for improving counter performance. Originally, a small high-performance counter was used to divide an incoming clock, thus providing a slower clock to a larger, lower-performance counter. This technique has since been adapted to synchronous counters.

In a synchronous counter, the first few bits of the counter are decoded to create a parallel Count Enable (CEP). This clock enable is used to reduce the effective clock rate. The carry chain in the more significant bits is, thereby, allowed several clock periods in which to settle. However, using this technique results in a counter that, without further adaptation, is non-loadable.

Typically, in the LCA implementation of such a counter, the critical delay is the generation and distribution of CEP. This delay can be shortened by pipelining CEP and using a high-speed Longline for its distribution. However, where ultimate speed is the objective, even the relatively small Longline delay can be eliminated.

To eliminate this delay, the LSB of the counter is replicated to create an "active Longline." This involves locating an LSB replica immediately adjacent to each bit in the counter. In counter organizations where one CLB provides the flip-flops for two counter bits, the number of replicas required is approximately half the number of bits in the counter.

In XC3000 designs, direct interconnect can be used between the LSB replicas and the counter bits. This results in an effective distribution delay of zero. In XC4000 designs, the residual routing delay is minimal.

Implementation

XC3000

The XC3000 design for the ultra-fast counter is shown in Figure 1. This design uses two parallel count enable signals, Q_0 and CEP2. Q_0 acts as a 1-bit prescaler, halving the effective clock rate in the rest of the counter. It is the distribution of Q_0 that is critical, and depends upon replication.

Even with the effective clock rate halved, it is necessary to use a second 2-bit prescaler for any significant length of counter. The parallel count enable signal (CEP2), generated by this second prescaler, occurs once every eight clock cycles. Reducing the effective clock rate by a factor of eight permits the use of a simple ripple carry scheme for the remaining bits of the counter. The Q_0 prescaler allows two clock cycles for the distribution of CEP2, and a Longline is adequately fast.

Except for the Q_1 flip-flop, the column of CLBs on the left consists entirely of replicated LSBs. Only one flip-flop, at the top of the column, is configured to toggle. The remaining flip-flops in the column act as slaves to this one master flip-flop.

These slave flip-flops are organized as a shift register with inverters between stages. At each stage there is a pair of flip-flops ($QX0_i$ and $QY0_i$) contained within a single CLB. The two flip-flops operate in parallel. This duplication permits both vertical direct interconnect to the next stage, and horizontal direct interconnect to the counter bits.

The first stage toggles by continuously loading the inverse of its current state. Stage two loads the inverse

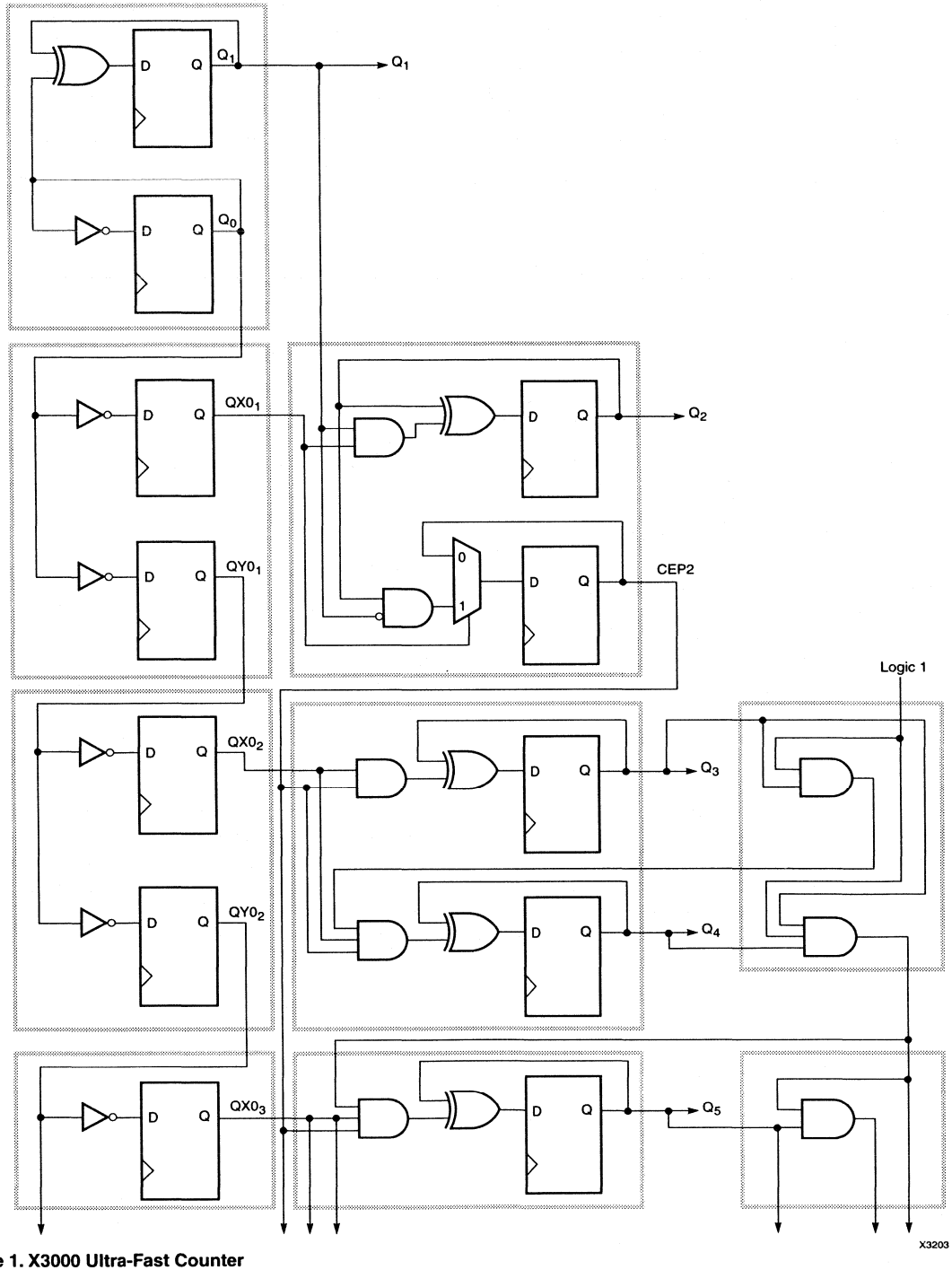


Figure 1. X3000 Ultra-Fast Counter

of stage one, delayed by one clock period. Given that stage one is toggling, this combination of inversion and delay causes stage two to operate in synchronism with stage one, as shown in Figure 2. Similarly, stage three operates in synchronism with stage two, and so on. This slave mode of operation guarantees that all N stages will operate in synchronism after no more than N-1 clocks, regardless of their initial state.

To avoid unnecessary loading on the direct interconnects, the Q0 output is taken from the last stage of the shift register. Otherwise, the additional loading would cause a small increase (~0.1 ns) in the direct interconnect delay, and this would reduce the maximum clock frequency by ~1 MHz.

The second prescaler, Q_1 and Q_2, is a simple 2-bit counter, enabled by Q_0. CEP2 is High for two clock periods while Q_1 and Q_2 are both High. The CEP2 pipeline flip-flop is also enabled by Q_0. In this way, CEP2 changes at the same time as Q_1 and Q_2, and each has two clock periods in which to set up. CLB input constraints require that Q_2 be externally routed to the CEP2 decoder.

The remaining bits of the counter use a ripple-carry scheme. Pairs of bits are implemented together, using two CLBs per pair. One CLB provides the two flip-flops, and is placed adjacent to a Q_0 CLB to exploit the direct interconnect. The second CLB implements the carry chain, with each pair of bits adding one T_{ILO} delay. To minimize the cumulative delay and maximize the counter length, direct interconnect should also be used in the carry path.

With all critical delays reduced to a clock-to-output delay plus a set-up time, with no routing delay, the minimum clock period is 10.5 ns (95 MHz). The ripple-carry delay in the more significant bits in an XC3000-125 counter is approximately 15 ns plus 5.7 ns per bit-pair. With the counter running at its minimum clock period, the carry chain has 84 ns in which to settle. This will permit up to 12 bit-pairs in the ripple carry path. A counter running at the maximum speed can, therefore, have up to 27 bits including the prescalers.

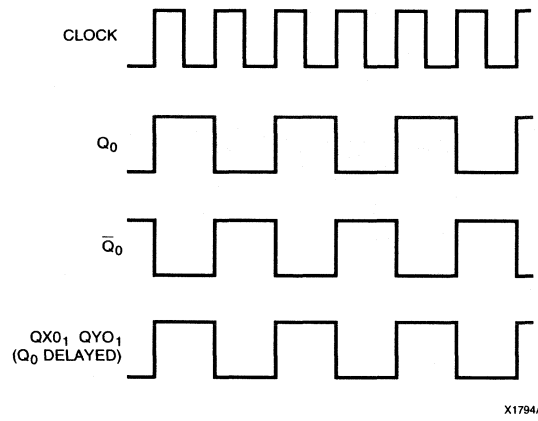


Figure 2. Operation of LSB Shift Register

XC4000

The XC4000 design, shown in Figure 3, is very similar to the XC3000 design. The principle difference is that the dedicated carry logic can be used in the more significant bits of the counter.

To maximize the performance, all critical paths are restricted to single-length interconnects, only one of which is driven from any output. This again requires that pairs of flip-flops be used in each stage of the LSB shift register. Using double-length interconnects or driving multiple single-length lines, the number of flip-flops can be reduced, with only a slight loss of performance.

The minimum clock period is the clock-to-output delay plus routing delay and set-up time. With the interconnection strategy described above, this can be kept below 9 ns (111 MHz). The ripple-carry delay in the more significant bits is 13 ns plus 1.5 ns per bit-pair. The 72 ns available permits a theoretical maximum counter length of 87 bits. In practice, the number of bits will be limited by the loading on the Long line distributing CEP2. The available time should allow counters in excess of 20 bits long to be constructed.

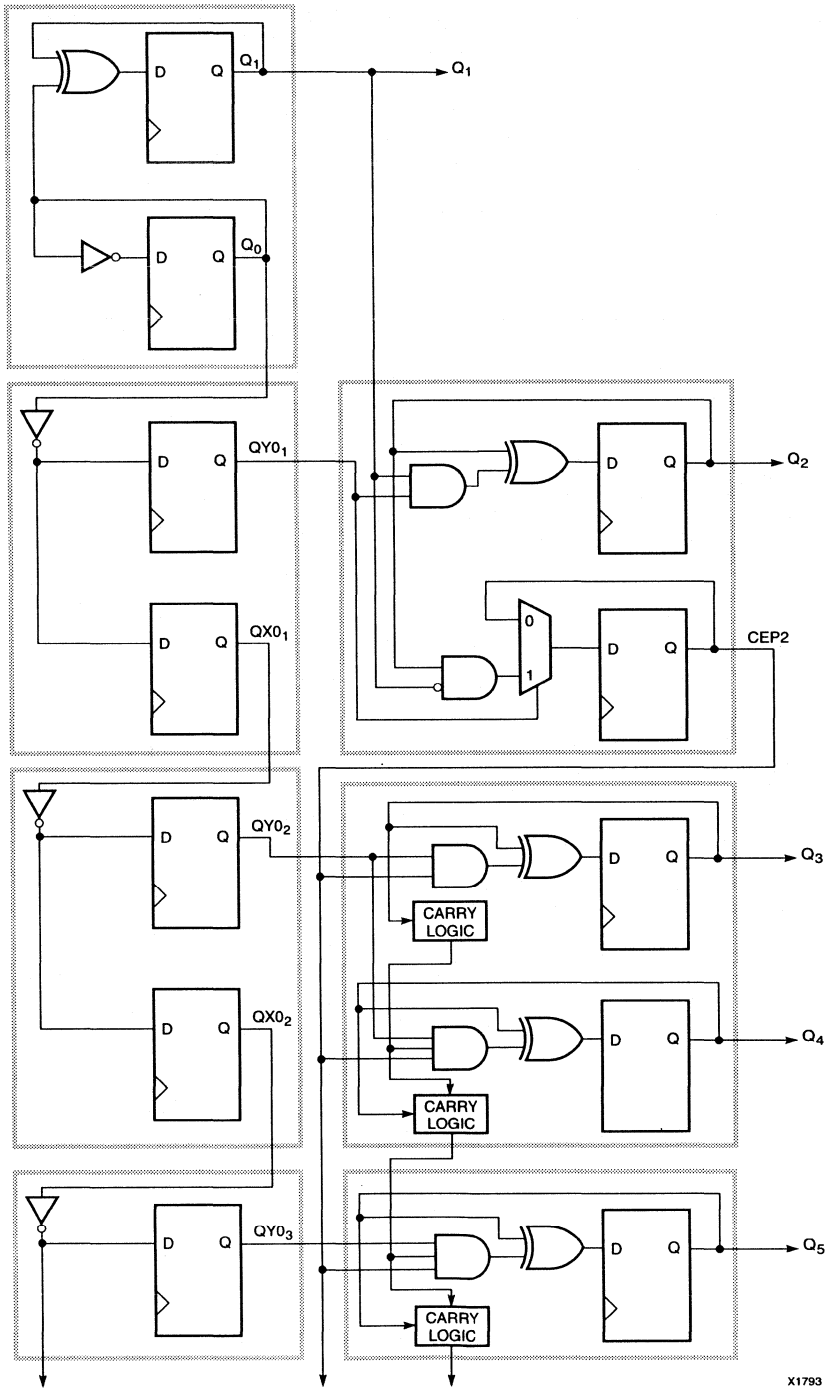


Figure 3. XC4000 Ultra-Fast Counter

Accelerating Loadable Counters in XC4000

XAPP 023.001

Application Note By BERNIE NEW

Summary

The XC4000 dedicated carry logic provides for very compact, high-performance counters. This Application Note describes a technique for increasing the performance of these counters using minimum additional logic. Using this technique, the counters remain loadable.

Xilinx Family

XC4000

Demonstrates

Dedicated Carry Logic
High-performance Counter Design

Introduction

The dedicated carry logic in XC4000 LCA devices provides a mechanism for very fast and efficient counters. While the ripple-carry scheme appears simplistic, the hardware implementation of the dedicated carry logic is very fast, and requires few CLBs. In fact, the implementation is so efficient that it defeats most attempts to replace it. It is possible, however, to augment the operation of the carry logic and obtain higher performance.

To reduce the ripple-carry delay, the effective length of the carry path must be shortened. This is achieved by dividing the counter into two sections that settle in parallel, as shown in Figure 1. The carry output of the less-significant section provides a parallel Count Enable (CEP) to the more-significant section.

The carry delay is reduced to the settling time of the more significant section, or the settling time of the less significant section plus the subsequent routing and count-enable times, whichever is greater. For optimum performance, these times should be balanced, requiring that the counter be divided into two unequal parts.

The use of CEP does not imply that these are prescaler techniques. In a prescaler counter, CEP is typically decoded from the least significant two or three bits. The CEP signal is then used to enable the remaining bits, such that their effective clock rate is one fourth or one eighth of the actual clock rate. This allows multiple clock periods for the remaining bits to settle, and the whole counter can be operated at the speed of the prescaler.

Using the prescaler technique, it is not possible to load the counter and guarantee that it will count correctly on the following clock cycle. The carry chain in the more significant bits is designed to settle in multiple clock periods. If the loaded data causes these bits to be enabled on the clock following the load operation, the carry path will not, in general, have had adequate settling time. Depending on the value loaded, it might not be possible to resume counting for several clock periods after the load operation.

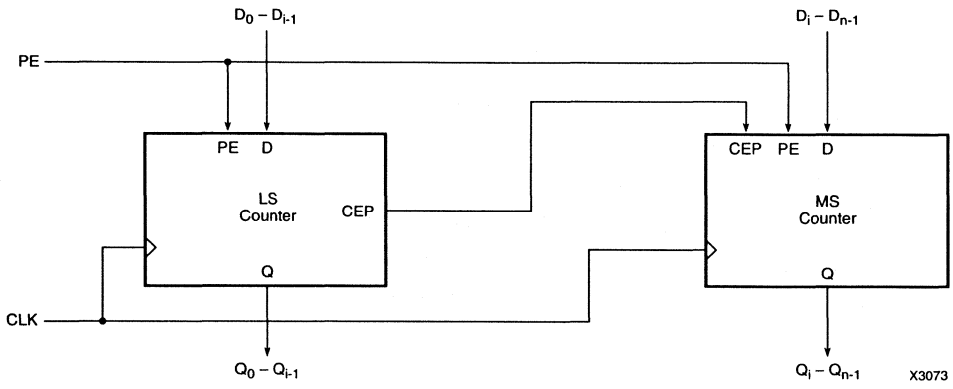


Figure 1. Accelerated N-Bit Counter

The acceleration technique described in this Application Note does not depend upon carry chains having multiple clock periods in which to settle; the entire carry chain settles within one clock period. However, the clock period is reduced because parallelism is introduced into the carry chain. The improvement is not as dramatic as with a prescaler, but loadability is retained.

Two versions of the technique are described below. One version uses two dedicated carry-logic chains, and is increasingly effective in longer counters. For shorter counters, a second version uses CLBs for the less significant section, and decreases the clock period by a fixed amount (1.5 ns in an XC4000-5). While the benefit from this second version is small, it can sometimes be crucial. Figure 2 illustrates the benefits derived from the two versions. In either case, one additional CLB is required to accelerate the counter.

Operating Description

Long-Counter Version

To accelerate long counters, the carry chain must be divided into two unequal parts. The less significant section should be shorter to accommodate the distribution and set-up times of CEP. For optimum performance, each section of the counter should contain an odd number of bits.

If the counter length is an exact multiple of four, the more-significant section should be 10 bits longer than the less-significant section. A 32-bit counter, for example, should be split into sections of 11 and 21 bits.

This split creates a 7.5-ns difference in settling times to accommodate the additional delay. The set-up time is 4 ns, and consequently, 3.5 ns is available for routing. A Longline should easily meet this requirement, leaving the speed controlled by the more-significant section of the counter.

As described in the Application Note, Estimating the Performance of XC4000 Adders and Counters (XAPP 018), the estimated minimum clock period for an N-bit counter is the following.

$$t_{CLK\text{-}CLK} = 13 + 0.75N \text{ ns}$$

Assuming that the speed of the accelerated counter is determined by the more-significant section, this reduces to the following.

$$t_{CLK\text{-}CLK} = 17.5 + 0.375N \text{ ns}$$

As a result, the clock period of a 32-bit counter is reduced from 37 ns to 29.5 ns.

For counters with an even length that is not divisible by four, the more-significant section should contain eight

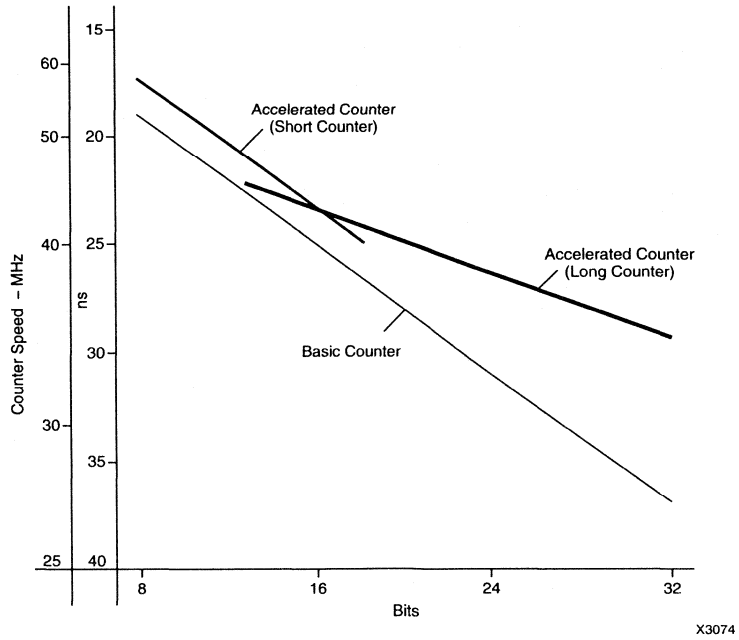


Figure 2. Counter Speed Comparison (Max Speed vs Counter Length)

more bits than the less-significant section. In this case, the speed of the counter will be controlled by its less significant section plus the additional CEP delays. While the minimum clock period is no longer as well-defined, it is again approximated by the above formula.

Splitting the counter into odd-length sections, one function generator is available in each section. As shown in Figure 3, these function generators can be used to generate CEP and Terminal Count (TC). To permit this, they should be G function generators, and share CLBs with the MSBs of each section.

The CEP signal uses CLB Enable Clock pins to control counting in the more significant section. Consequently, it must be forced to a one while the counter is being loaded. CEP is, therefore, defined as C_{OUT0} + PE.

The carry input to the more-significant section of the counter is forced to a one, and the carry chain in this section is independent of the less significant bits. In order for TC to reflect the state of the entire counter, it must be generated as MSCOUNT0 • LSCOUNT0.

One benefit of this counter is that TC is available without additional time delay or CLB cost. The CLB count of the accelerated counter matches that of the unaccelerated counter if TC is generated. If TC is not required, the unaccelerated counter can be one CLB smaller.

Short-Counter Version

For counters shorter than 16 bits, the following design should be used. It is based on the same fundamental approach as the counter described above, but offers greater benefit in short counters.

As shown in Figure 4, the less significant section of the counter is two bits long, and is implemented using function generators instead of the dedicated carry logic. The more significant section of the counter is N-2 bits long and is implemented using the carry logic.

As in the previous design, CEP is forced to a one while the counter is loaded. This permits the enable clock pin to be used as Count Enable with the Parallel Enable taking priority.

The 1.5-ns performance advantage requires that the counter speed be dominated by the more significant section, which is two bits shorter than the unaccelerated counter and, therefore, faster. With good routing, this requirement can be met in counters of six or more bits.

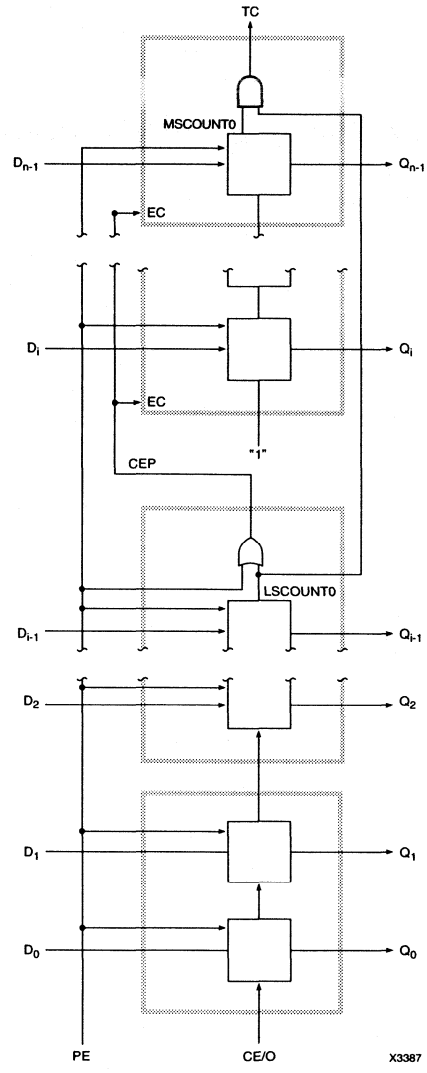


Figure 3. Long Accelerated Counter

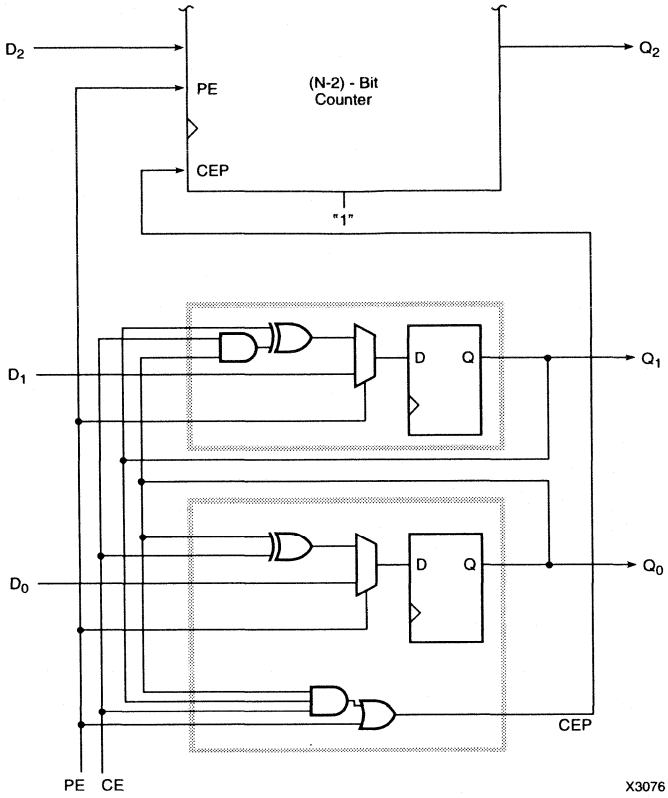


Figure 4. Short Accelerated Counter

Complex Full-Featured Counters Run at 83 MHz

XAPP 034.002

Application Note By JEFFREY GOLDBERG

Summary

This Application Note illustrates the implementation of long high-speed counters in Xilinx EPLDs. The Universal Interconnect Matrix eliminates the speed degradation usually associated with increasing counter length.

Xilinx Family

XC7200/XC7300

Demonstrates

High-speed Counter Design

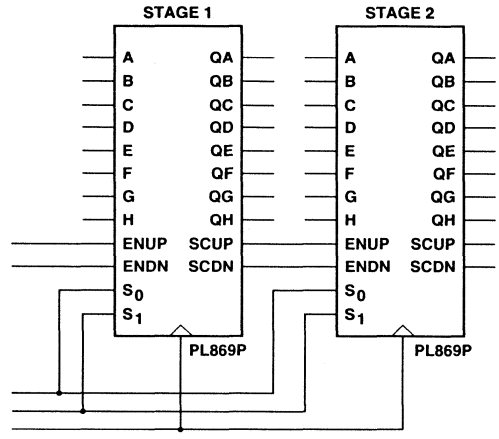
Introduction

Instrumentation, video/graphics and digital-signal-processing (DSP) applications use complex full-featured counters as building blocks. These counters might be used for prescalers, pulse generators, frequency counters or complex address generators. Typical counter requirements are listed below.

- Up-Down Operation
- Synchronous Load
- Synchronous Clear
- Cascadable to 32 Bits at 83.3 MHz

The Xilinx family of EPLDs is well suited for implementing high-speed full-featured counters. The wide fan-in of each Function Block and the Macrocell XOR gate, coupled with the capabilities of the Universal Interconnect Matrix (UIM™), permit high-speed counters with no trade-off between speed and density. Any counter output can feed any other Macrocell in the device with no speed degradation thus simplifying real-world systems.

High-speed counters are easily implemented by cascading up to four instances of the 8-bit counter (PL869P) from the component library. Each 8-bit counter occupies nine Macrocells. Any number of 8-bit counters can be cascaded up to the capacity of the device, yet the 83.3 MHZ maximum speed is maintained for any length of counter. Figure 1 illustrates how counters can be cascaded. In this example, two 8-bit counters are cascaded into a 16-bit counter.



Functional Truth Table

S_1	S_0	ENUP	ENDN	FUNCTION
L	L	X	X	CLEAR
L	H	L	L	HOLD
L	H	H	L	COUNT UP
L	H	L	H	COUNT DOWN
L	H	H	H	HOLD
H	L	X	X	LOAD
H	H	X	X	HOLD

X1795

Figure 1. Cascaded Counter

Counter Implementation

Figure 2 shows one bit of the PL869P counter. The XOR gate reduces the required number of product terms to four per bit. The remaining 13 product terms are not shown. The Macrocell output is available through the Universal Interconnect Matrix (UIM) to all Macrocells in the device. In an XC7354-10, the clock-to-setup time from any Macrocell flip-flop through the UIM to any Macrocell flip-flop is 12 ns. This permits a maximum clock frequency of 83.3 MHz. Macrocell outputs are also available as device outputs.

The key to implementing long, complex, high-speed counters is the UIM ability to simultaneously provide interconnect and logic function. Generating product-terms in the UIM reduces the number of inputs required in each Function Block, and improves speed and density. AND operations performed in the UIM cause no additional delay between Function Blocks. The resulting advantages are demonstrated by the PL869P counter.

The PL869P counter is actually composed of three elements, as shown in Figure 3. Two of these elements, the logic for the eight counter bits and a carry lookahead for the down-counting serial carry (SCDNX) are implemented in Macrocells. The serial carry outputs (SCUP and SCDN) for each stage of the counter, however, are formed in the UIM.

While it appears that the carry signals ripple through the AND gates of each stage, the carry outputs are actually generated in parallel. The critical path limiting the maximum count frequency is the propagation delay from the carry output of the least significant eight bits to the count-enable input of the most significant eight bits. This delay is constant, independent of the counter length.

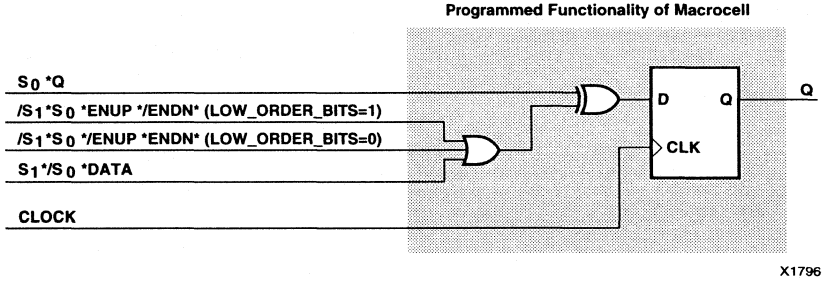


Figure 2. PL869 Counter Bit

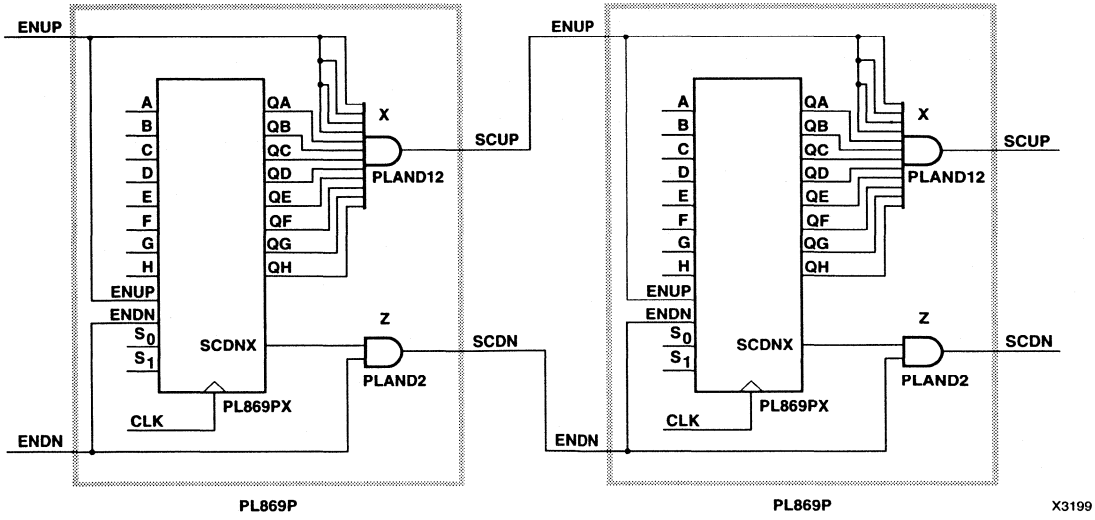


Figure 3. PL869P Counter

High Performance Counters Using Xilinx EPLDs with ABEL-HDL

XAPP 038.001

Application Note By JEFFREY GOLDBERG

Summary

Xilinx EPLDs are capable of implementing counters that operate at the maximum device frequency. This Application Note explains how ABEL-HDL can be used to implement such counters.

Xilinx Family	*Demonstrates*
XC7200/XC7300	High-speed Counter Design

Introduction

The Xilinx XC7200 and XC7300 families of EPLD devices are well suited for implementing large, fully featured, high-speed counters. Such counters benefit from the wide fan-in of the Function Blocks (21 inputs) and Macrocells (17 product terms), the XOR gate in the Macrocell, and the logic capabilities of the Universal Interconnect Matrix (UIM). Together, these features permit the implementation of high-speed counters with no trade-off between speed and density.

This Application Note demonstrates the use of ABEL-HDL and the ABEL XFER utility to generate counter equations. ABEL-HDL is a powerful tool for describing high-performance counters. Just as with the more familiar 22V10 low-density PAL device, only minimal knowledge of the Xilinx EPLD Function Block is required when generating the source code. Properly used, ABEL-HDL produces PLUSASM equations that map efficiently into the Xilinx EPLD architecture, and the resulting equations implement long, complex counters that run at the full speed of the device.

Architecture Overview

A rudimentary understanding of the Xilinx EPLD architecture is helpful when designing efficient counters. Figure 1 shows the three major components of the architecture.

The Function Blocks are PAL-like logic blocks, where most, but not all, logic functions are performed. Each Function Block contains nine Macrocells. These Macrocells share 21 inputs, and every Macrocell has its own output. In this application, each Macrocell implements one bit of the counter.

Interconnection among Function Blocks is provided by the UIM. This fully populated switch matrix connects all device inputs and Function Block outputs to all the inputs of every Function Block. In addition, the UIM can perform an ANDs of all Function Block outputs and device inputs.

I/O blocks interface the Function Blocks to the device pins. They contain input latches and registers that can be useful for data storage.

The keys to implementing long, complex, high-speed counters in Xilinx EPLDs are the XOR gate in each Macrocell and the AND capability of the UIM. The XOR gate is particularly useful when implementing loadable counters, Figure 2. The XOR gate reduces the required number of product terms to only four per bit; 13 unused product terms remain available.

Loadable counters can be implemented more efficiently with XOR gates and D-type flip-flops than with T-type flip-flops. The Xilinx EPLD architecture can force the counter feedback Low whenever LOAD is asserted. The counter bit can then be loaded using only one product term. Counters implemented with T-type flip-flops require two product terms for synchronous loading, since the conditions that force the flip-flop to toggle depend on both the data input and the counter state.

This difference is especially important when the counter is loaded from multiple sources. As shown in Figure 3, the Xilinx EPLD requires only one additional product term for each additional source, while a T-type flip-flop would require two additional product terms.

The UIM is actually a very wide AND array that operates without introducing additional delay. Implementing product terms in the UIM reduces the number of inputs into each Function Block, and also improves speed and density. Counters implemented using the UIM can run at the full device speed, independent of their length.

Figure 4 shows the implementation of a 27-bit counter. Nine counter bits are mapped into each Function Block. Notice how the outputs from the first nine bits (Q0...8) and

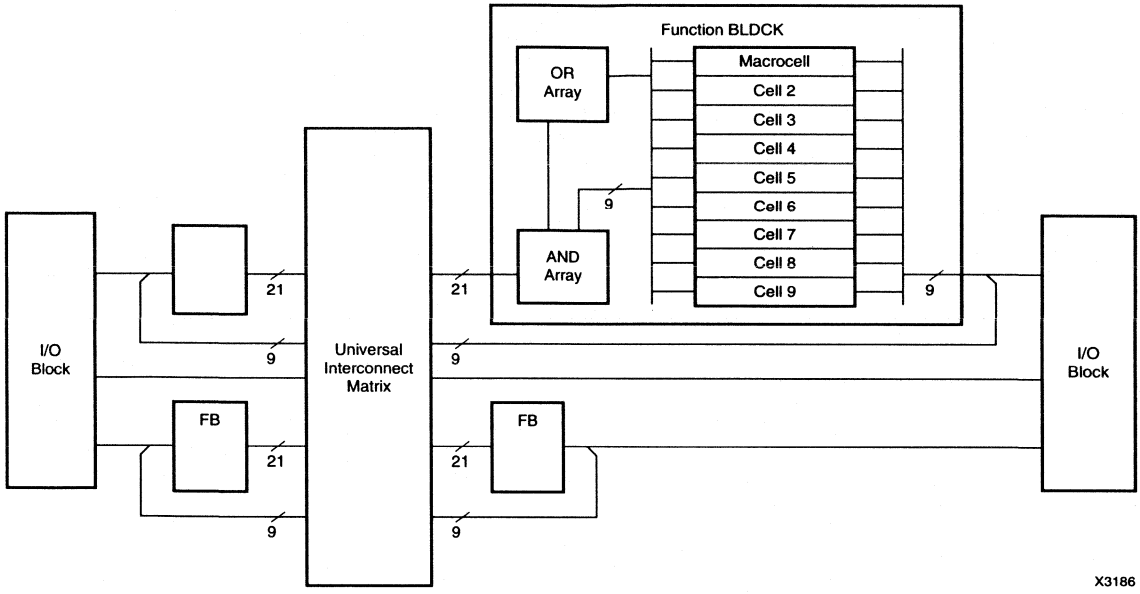


Figure 1. Xilinx EPLD Architecture

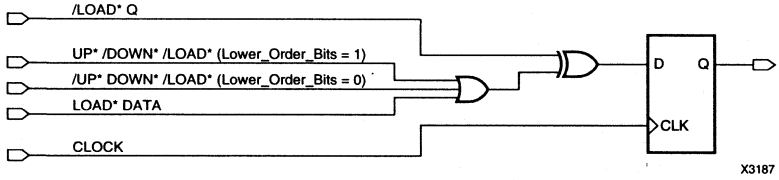


Figure 2. Bidirectional, Loadable Counter Bit

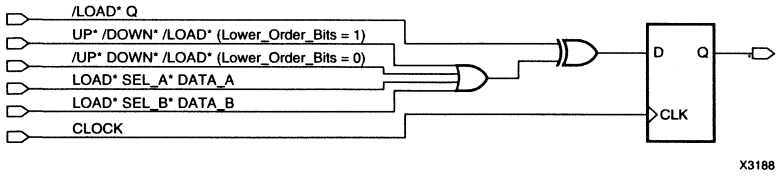


Figure 3a. Loadable Counter with Multiplexed Inputs, Xilinx EPLD Architecture

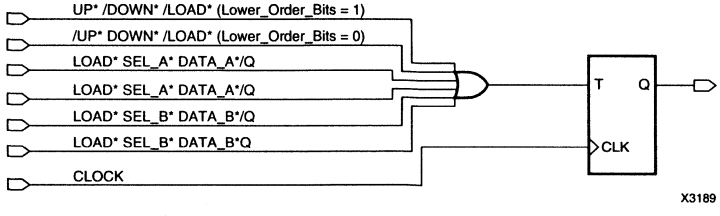


Figure 3b. Loadable Counter with Multiplexed Inputs. T-type Flip-Flop Architecture

the Count Enable are ANDed in the UIM to form their Terminal Count, COUNT_FB2. This Terminal-Count signal enables the second group of nine bits (Q9...17). Similarly, most significant of nine bits are enabled by Terminal Count, COUNT_FB3, generated by ANDing the second group of nine outputs with the least significant nine and the Count-Enable signal.

The critical path that limits the count frequency runs from the least significant bit to the Count-Enable input of the most significant stage. This path delay is independent of the number of function blocks it spans. Consequently, any length counter that spans multiple Function Blocks can operate at f_{MAX}, 60 MHz in an XC7236A-16.

ABEL-HDL Counter Implementation

When generating ABEL-HDL counter descriptions for a Xilinx EPLD, keep in mind the basic features of the architecture.

- Each Function Block has 21 inputs, and comprises nine Macrocells with one output each.
- Each Macrocell has a D-type flip-flop preceded by an XOR gate.
- The UIM is a wide AND array

For the highest performance, maximize the number of counter bits in each Function Block. In unidirectional counters, nine bits of a loadable up counter or down counter can fit into a single Function Block. Bidirectional counters, however, can only fit eight bits into a Function Block. This reduction is caused by the need for two Terminal Count signals, one for up counting and one for down counting.

As shown in Figure 5, the up-counting Terminal-Count signal can be generated in the UIM, just as it would be in an up counter. The down-counting Terminal Count is the AND function of the inverted counter bits, i.e., an all-zero

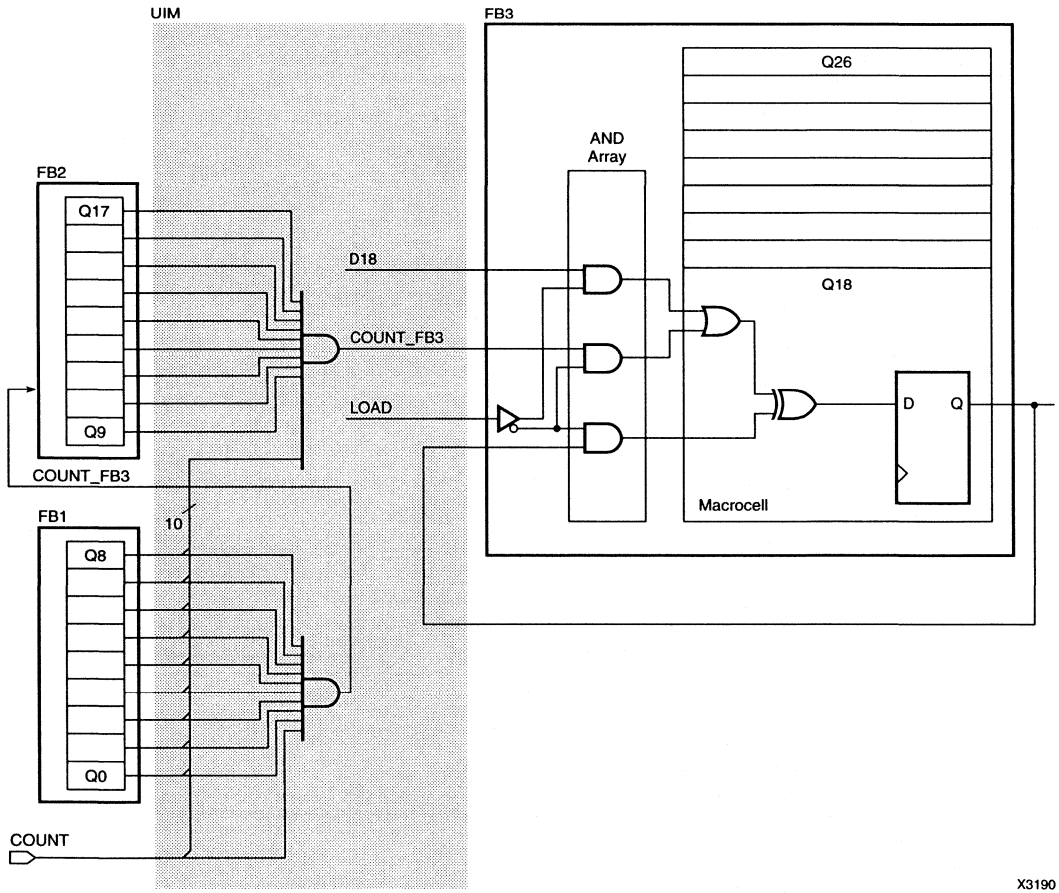


Figure 4. 27-Bit Loadable Up Counter

detect in place of an all-one detect. This requires the use of a Macrocell, and consequently, only eight are available for counter bits.

To avoid additional logic delay, the down-counting Terminal-Count signal is pipelined; the state immediately preceding Terminal Count is detected, and the result is registered on the same clock that moves the counter into its Terminal-Count state. To ensure correct operation during loading, the load value is inspected, and the pipeline flip-flop set appropriately at the load clock.

If the counter outputs must be compared to a dynamic value, leave one Macrocell available in each Function Block to perform the comparison. The fitter portion in the ABEL complier recognizes that the comparator and counter can share common Function Block inputs, and maps them into the same Function Block.

Once the number of counter bits that fit into any Function Block is determined, the ABEL source code can be written. Here are some key points to remember when generating the source code.

- Declare each counter bit with the istype 'reg,xor' attribute to take full advantage of the XOR functionality of the Macrocell ALU. Then use the xor_factors keyword to define the signal that drives one input of the XOR gate. If at all possible, keep this signal down to one product term.
- Declare the terminal count look-ahead function of each stage, e.g., the down terminal count of bidirectional counters, as node istype 'reg'. This function will be mapped into a Macrocell.
- Declare the up-terminal count, or a down-counter down-terminal count, of each stage as node istype 'com'. Since the Function Block outputs and inputs can be inverted, modulo-n stages can be defined and the terminal count can still be generated in the UIM. This function will be mapped into the UIM when it is declared as NODE (UIM) in the PLUSASM top-level design file.
- While implementing counters, be sure to completely specify the function so that ABEL can minimize the logic. Whenever PLUSASM is generated for a function declared with an xor attribute, ABEL expresses the equation in PLUSASM ALU syntax. XEPLD will not further minimize these equations.
- Unless a counter is manually partitioned in the top level design file using PLUSASM 'Partition' statements, the fitter is free to map the counter bits in any way it chooses. Consequently, the counter may map into multiple Function Blocks, achieving what the fitter considers a best fit given the I/O requirements; the fitter may not map the maximum number of counter bits into each Function Block.

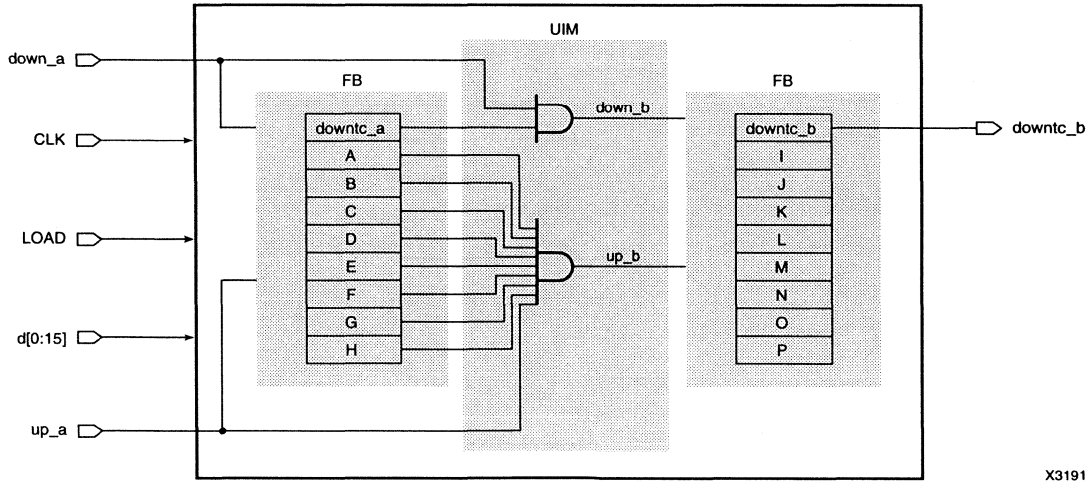


Figure 5.16-Bit Bidirectional Loadable Counter

Additionally, unnecessary inputs may be consumed in the Function Blocks that contain the higher order counter bits. These inputs are consumed by lower order counter bits that are ANDed in the Function-Block AND array to form Terminal Count signals, instead of being ANDed in the UIM.

This may not be a problem. Provided a design can access all the Macrocells it requires, any additional inputs are available at no cost. When it is necessary to free up Function Block inputs, first determine from the fitter mapping report how the counter is mapped. Using this information, modify the ABEL-HDL source code to create counter block that correspond to the counter bits that are mapped into each Function Block. Terminal Count signals can then be generated using UIM ANDs, thus minimizing the number of inputs used on each Function Block.

In practice, the fitter tends to keep many of the counter bits together. The best strategy is to assume that the maximum number of bits will be implemented in each Function Block, and then optimize the source code to free up more inputs if necessary.

The following examples show how to implement counters in Xilinx EPLDs using ABEL-HDL. Example 1 implements the simple 27-bit loadable up-counter shown in Figure 4. Example 2 implements an 8-bit loadable, bidirectional counter, as shown in Figure 2. Example 3 expands upon Example 2, and uses ABEL-HDL to implement the 16-bit loadable, bidirectional counter shown in Figure 5. The ABEL source codes and PLUSASM top level design files follow.

Example 1. 27-Bit Loadable Up Counter

ABEL Source Code

```
module upcntr
 title 'loadable 27 bit loadable up counter
       each 9 bit stage fits in one fb
       up terminal counts formed in uim
       Jeffrey Goldberg
       Xilinx';
upcntr device;
" Inputs
      Clk,load,count                          pin;
      d0,d1,d2,d3,d4,d5,d6,d7,d8              pin;
      d9,d10,d11,d12,d13,d14,d15,d16,d17      pin;
      d18,d19,d20,d21,d22,d23,d24,d25,d26     pin;
" Outputs
     q0,q1,q2,q3,q4,q5,q6,q7,q8               pin istype 'reg,xor';
     q9,q10,q11,q12,q13,q14,q15,q16,q17       pin istype 'reg,xor';
     q18,q19,q20,q21,q22,q23,q24,q25,q26      pin istype 'reg,xor';
" Nodes
     count_fb2, count_fb3                    node istype 'com';
" Variables
     data_fb1  = [d8..d0];                   " data inputs
     data_fb2  = [d17..d9];
     data_fb3  = [d26..d18];
     fb1       = [q8..q0];                   " counter outputs
     fb2       = [q17..q9];
     fb3       = [q26..q18];
xor_factors  fb1 := fb1 & !load;             " q.d2 = q & !load
xor_factors  fb2 := fb2 & !load;
xor_factors  fb3 := fb3 & !load;
equations
" Function Block 1
    fb1 := (fb1 + 1) & count & !load         " count up
        # fb1 & !count & !load               " hold
        # data_fb1 & load;                   " load
" Function Block 2
    fb2 := (fb2 + 1) & count_fb2 & !load     "count up
        # fb2 & !count_fb2 & !load           "hold
        # data_fb2 & load;                   "load
" Function Block 3
    fb3 := (fb3 + 1) & count_fb3 & !load     "count up
        # fb3 & !count_fb3 & !load           "hold
        # data_fb3 & load;                   "load
" Form count enables in uim
       count_fb2 = count & (fb1==511);
        count_fb3 = count & (fb1==511) & (fb2==511);
end
```

PLUSASM Top Level Design File, COUNTER1.PLD

```
TITLE          COUNTER1
AUTHOR         JEFFREY GOLDBERG
COMPANY        XILINX
DATE           3/2/93
INCLUDE_EQN 'UPCNTR.PLD'
CHIP    COUNTER1    XEPLD
INPUTPIN       LOAD COUNT D0 D1 D2 D3 D4 D5 D6 D7 D8 D9 D10
               D11 D12 D13 D14 D15 D16 D17 D18 D19 D20 D21
               D22 D23 D24 D25 D26
OUTPUTPIN      Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13
               Q14 Q15 Q16 Q17 Q18 Q19 Q20 Q21 Q22 Q23 Q24
               Q25 Q26
NODE (UIM)     COUNT_FB2 COUNT_FB3
FASTCLOCK      CLK
EQUATIONS
```

Example 2. 8-Bit Loadable, Bidirectional Counter

ABEL Source Code

```
module updncntr
 title 'loadable 8 bit loadable up/down counter
       up terminal count formed in uim
       down terminal count formed in macrocell
       Jeffrey Goldberg
       Xilinx';
updncntr device;
" Inputs
      Clk,load,up,down                       pin;
      d0,d1,d2,d3,d4,d5,d6,d7                pin;
" Outputs
     q0,q1,q2,q3,q4,q5,q6,q7                 pin istype 'reg,xor';
     downtc,done                            pin istype 'reg';
" Nodes
     uptc                                   node istype 'com';
" Variables
     data     = [d7..d0];                   " data inputs
     count    = [q7..q0];                   " counter outputs
xor_factors  count := count & !load;        " q.d2 = q & !load
equations
    count := (count + 1) & up & !down & !load     " count up
          # (count - 1) & !up & down & !load      " count down
          # count & up & down & !load             " hold
          # count & !up & !down & !load           " hold
          # data & load;                          " load
" Form down terminal count in macrocell
    downtc := (count == 1) & !up & down & !load   " count = 0 on next
clock
          # (count == 0) & up & down & !load      " holding at 0
          # (count == 0) & !up & !down & !load    " holding at 0
          # (data == 0) & load;                   " loading 0
" Form up terminal count in uim
        uptc = up & (count==255);
" Send uptc off-chip
        done := uptc;
end
```

PLUSASM Top Level Design File, UPDNCNT8.PLD

```
TITLE          UPDNCNT8
AUTHOR         JEFFREY GOLDBERG
COMPANY        XILINX
DATE           3/2/93
INCLUDE_EQN 'UPDNCNTR.PLD'
CHIP    UPDNCNT8    XEPLD
INPUTPIN       LOAD UP DOWN D0 D1 D2 D3 D4 D5 D6 D7
OUTPUTPIN      Q0 Q1 Q2 Q3 Q4 Q5 Q6 Q7 DOWNTC DONE
NODE (UIM)     UPTC
FASTCLOCK      CLK
EQUATIONS
```

Example 3. 16-Bit Loadable, Bidirectional Counter

ABEL Source Code

```
module abelcntr
 title 'loadable 16 bit up/down counter
       each 8 bit stage fits in one fb
       up terminal count formed in uim
       down terminal count lookahead generated in macrocell
       second stage up and down count enables formed in uim
       Jeffrey Goldberg
       Xilinx';
  abelcntr device;
" Inputs
      Clk,load,up_a,down_a                    pin;
      d0,d1,d2,d3,d4,d5,d6,d7                 pin;
      d8,d9,d10,d11,d12,d13,d14,d15           pin;
" Outputs
     downtc_b                                 pin  istype 'reg';
"Nodes
     a,b,c,d,e,f,g,h                          node istype
'reg,xor';
     i,j,k,l,m,n,o,p                          node istype
'reg,xor';
     up_b, down_b                             node istype 'com';
     downtc_a                                 node istype 'reg';
     data_a    = [d0..d7];                    " data inputs
     data_b    = [d8..d15];
     count_a   = [a,b,c,d,e,f,g,h];           " counter outputs
     count_b   = [i,j,k,l,m,n,o,p];
     uptc_a    = (count_a == 255);" terminal counts
     uptc_b    = (count_b == 255);
xor_factors  count_a := count_a & !load; count.d2 = count & !load
xor_factors  count_b := count_b & !load;
equations
    count_a := (count_a + 1) & up_a & !down_a & !load      " count up
          # (count_a - 1) & !up_a & down_a & !load         " count down
          # count_a & up_a & down_a & !load                " hold
          # count_a & !up_a & !down_a & !load              " hold
          # data_a & load;                                 " load
    downtc_a := (count_a == 1) & !up_a & down_a & !load " counting
                                                          down
          # (count_a == 0) & up_a & down_a & !load        " holding
                                                          counter= 0
          # (count_a == 0) & !up_a & !down_a & !load      " holding
                                                          counter= 0
          # (data_a  == 0) & load;                        " loading 0
" form count enables in uim
        up_b = uptc_a & up_a;
        down_b = downtc_a & down_a;
    count_b := (count_b +1) & up_b & !down_b & !load
          # (count_b -1) & !up_b & down_b & !load
          # count_b & up_b & down_b & !load
          # count_b & !up_b & !down_b & !load
          # data_b & load;
   downtc_b := (count_b == 1) & !up_b & down_b & !load
          # (count_b == 0) & up_b & down_b & !load
          # (count_b == 0) & !up_b & !down_b & load
          # (data_b  == 0) & load;
end
```

PLUSASM Top Level Design File, ABELCNT1.PLD

```
TITLE          ABELCNT1.PLD
AUTHOR         JEFFREY GOLDBERG
COMPANY        XILINX
DATE           3/2/93
INCLUDE_EQN 'ABELCNTR.PLD'
CHIP    ABELCNT1    XEPLD
INPUTPIN       LOAD UP_A DOWN_A D0 D1 D2 D3 D4 D5 D6 D7 D8
               D9 D10 D11 D12 D13 D14 D15
OUTPUTPIN      DOWNTC_A
NODE           A B C D E F G H I J K L M N O P DOWNTC_A
NODE (UIM)     UP_B DOWN_B
FASTCLOCK      CLK
EQUATIONS
```

High-Speed Custom Length Binary Counters

XAPP 040.001

Application Note By JEFFREY GOLDBERG

Summary

This Application Note describes how to use Xilinx EPLDs for high-speed, binary counters that run at the full rated speed of the device. These area-efficient, custom-length counters use standard 4- and 8-bit library components.

Xilinx Family

XC7200/XC7300

Demonstrates

High-Speed Counters

Introduction

High-performance XC7000 binary counters are easily designed in the XEPLD schematic-capture environment using standard library components. When properly cascaded, these components provide counters that operate at f_{MAX} of the EPLD, independent of the counter length and complexity.

The Xilinx EPLD component library contains three binary up-counters PL161, PL163 and PLCTR8/T. Individual bits of these counters are implemented in Function Blocks. Terminal Count signals, however, are implemented in the Universal Interconnect Matrix (UIM), shown as an AND gate in Figure 1.

Counter Design

Optimizing Terminal Count

Cascaded counters can run at f_{MAX}, if the Synchronous-Carry-Out signal (SCO) is generated in the UIM, Figure 2. However, since the UIM cannot drive an output directly, the operating frequency must be reduced if SCO is required off-the-chip in the same clock cycle. SCO must pass through an additional Macrocell, effectively halving the performance, unless a pipeline look-ahead method is utilized.

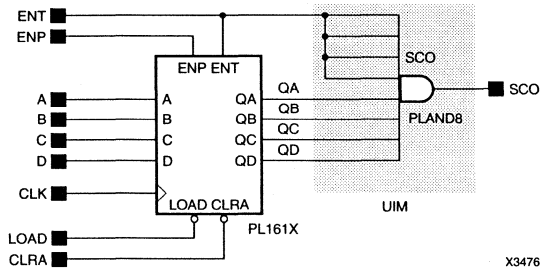


Figure 1. Typical 4-Bit Up-Counter Component

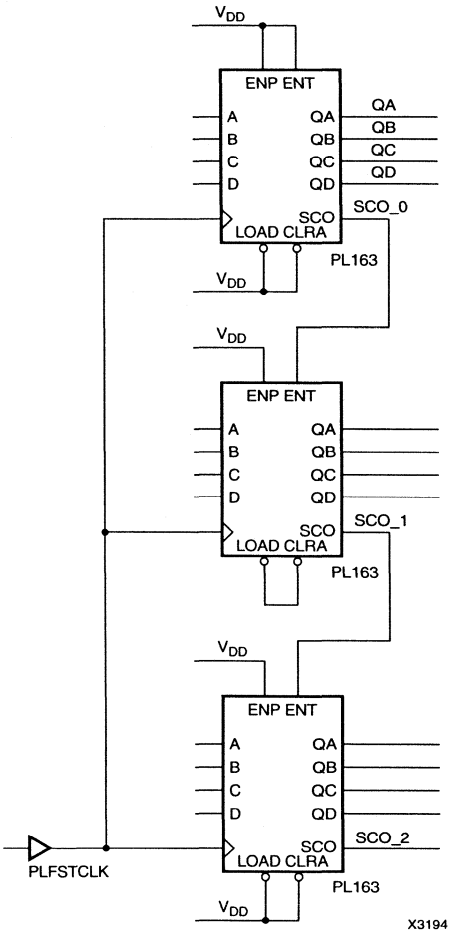


Figure 2. Typical Cascaded Counter

Figure 3 shows this technique that makes SCO available off-chip within the fMAX cycle time. A Count-Enable look-ahead circuit anticipates by one clock period when the counter will reach its terminal count, permitting the signal to be pipelined. Consequently, SCO is available to the output directly from the flip-flop for the duration of the Terminal-Count clock period, as shown in Figure 4.

Because the LSB of the least significant 4-bit counter is inverted for the look-ahead circuit, the SCO output of those four bits cannot be used without slowing the counter; instead, the pipelined SCO signal must be used. If the counter is to be loaded without restriction, the SCO pipeline flip-flop must also be loaded.

Customizing Counter Length

Custom-length counters using standard 4- and 8-bit library components often leave unused outputs in the design. Floating outputs are removed by the XEPLD software to reclaim the unused macrocells.

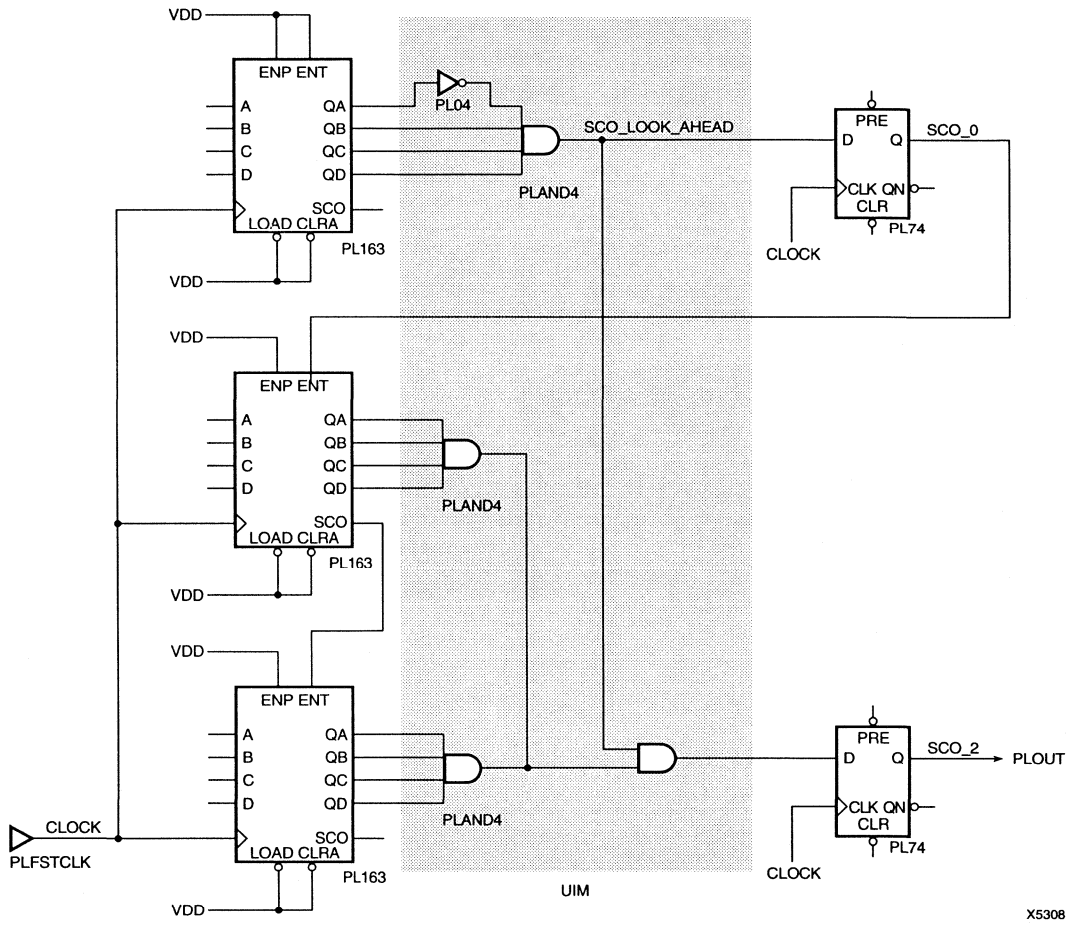


Figure 3. Cascaded Counter with Pipelined SCO

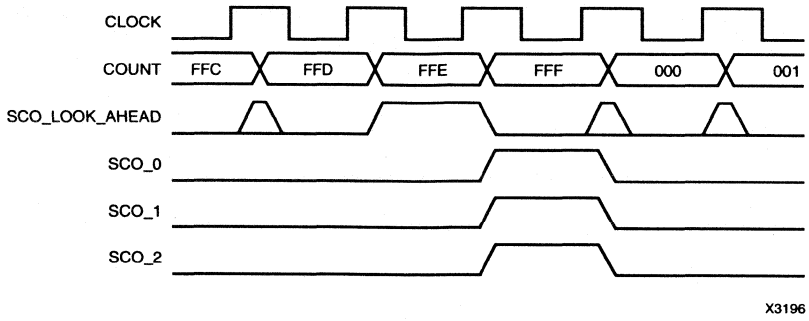


Figure 4. SCO Waveform Diagram

Adders, Subtracters and Accumulators in XC3000

XAPP 022.000

Application Note By PETER ALFKE and BERNIE NEW

Summary

This Application Note surveys the different adder techniques that are available for XC3000 designs. Examples are shown, and a speed/size comparison is made.

Xilinx Family

XC3000/XC3100

Demonstrates

Adder Techniques

Introduction

There are many ways to implement binary adders, subtracters and accumulators in LCA devices. Various approaches offer different trade-offs between size and speed.

Most compact, but slowest, is a bit-serial technique that operates on one or two bits per clock cycle, generating sum and carry. The sum is fed to an output shift register; the carry is stored and used in the subsequent bit time.

The most compact combinatorial (parallel) adder, subtracter, or accumulator consists of cascaded CLBs. Each CLB implements a full adder, accepting one bit of each operand and an incoming carry. The CLB generates the sum and an outgoing carry. A 16-bit function is completed in 16 CLB delays, and requires 16 CLBs.

With its 5-input function generator, an XC3000 CLB can implement additions two bits at a time. Three CLBs can each handle two input bits of each operand and an input carry to generate the two sum outputs and an outgoing carry. A 16-bit function requires 24 CLBs but the operation is completed in eight CLB delays.

For faster operation, a look-ahead carry technique can be used. Made popular by the 74181 ALU and its descendents, look-ahead carry uses Carry Propagate and Carry Generate signals to reduce the ripple-carry delay. Using look-ahead carry techniques in the XC3000, a 16-bit addition can be completed in five CLB delays, using 30 CLBs.

An even faster conditional-sum algorithm was originally described by J. Sklansky. Using this algorithm, a 16-bit adder requires 41 CLBs, but settles in only three CLB delays. With careful layout, the propagation delay through such an adder can be less than 20 ns in an XC3100-3.

Note that all Xilinx adder structures can be used as accumulators with no size penalty. Unlike conventional gate arrays and similar structures, LCA devices provide dedicated flip-flops in each CLB that can be used for the accumulator register. Since the flip-flop set-up time through the function generator usually matches the combinatorial propagation delay of the CLB, the set-up time for accumulator operands is similar to the propagation delay of the adder.

Bit-Serial Adders

The CLB architecture is ideally suited for bit-serial arithmetic. As shown in Figure 1, the two operands are serialized in shift registers, and presented, LSB first, to the serial arithmetic unit. The sum is created as a serial bit stream, again LSB first, that is converted to parallel data in a third shift register. Alternatively, one of the input shift registers may serve as the output register, with the sum shifted in to replace the operand.

The arithmetic unit, Figure 2, comprises a 1-bit full adder/subtracter and a carry/borrow flip-flop, and can be implemented in a single CLB. Before commencing an operation (addition or subtraction) the carry/borrow flip-flop must be cleared. Subsequently, sum or differences are passed to the output shift register, while carries or borrows are stored for inclusion in the next bit of the serial operation.

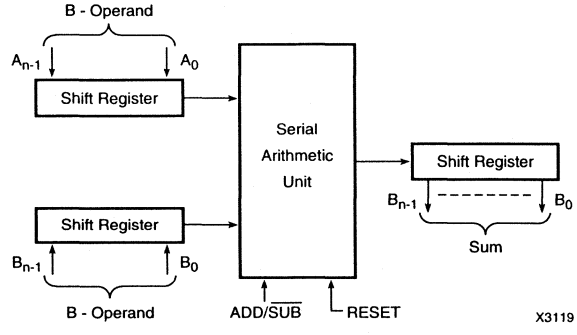


Figure 1. Serial Bit Adder/Subtracter

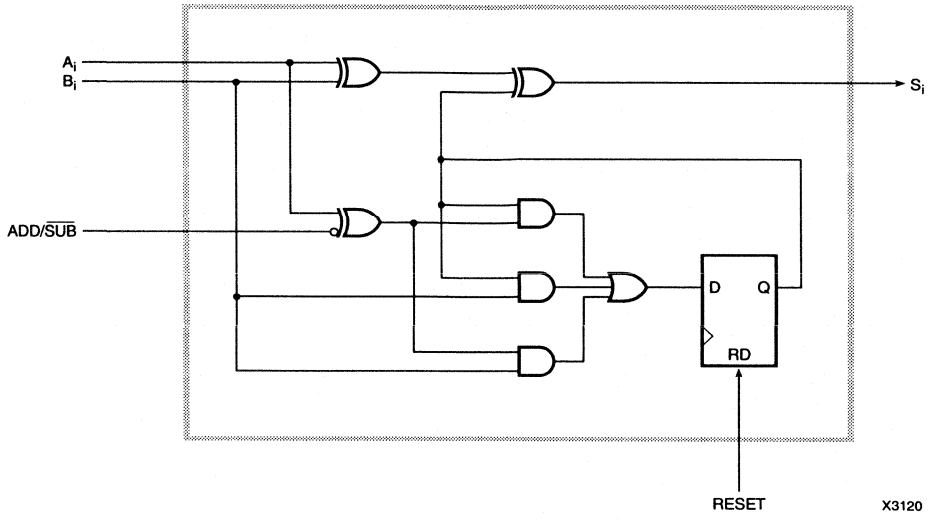


Figure 2. Serial Arithmetic Unit

While the number of clocks required to complete the operation equals the number of bits, the clock period can be very small because of the shallow logic. For maximum clock speed, the first bit of the output shift register should be implemented in the same CLB as the arithmetic unit.

Faster bit-serial operation can be obtained by simultaneously operating on two bits, Figure 3. Odd and even bits of each operand are loaded into separate shift registers. The arithmetic unit takes in two bits of each operand, and produces two sum bits per clock. These sum bits are loaded into odd and even output shift registers.

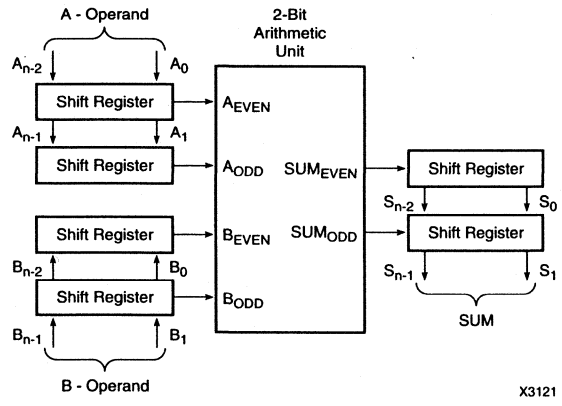


Figure 3. 2-Bit Serial Adder

Figure 4 shows the 2-bit arithmetic unit. Both sum bits are derived in parallel, and a single carry is generated and stored for the next cycle. This arithmetic unit permits adders and subtracters to be constructed, but not adders/subtracters. For adders/subtracter operation, the arithmetic unit should implement an adder; to generate A–B, the A-operand should be inverted while loading the operand shift register, and the sum bits should be inverted into the output register. The carry flip-flop is cleared before each operation, regardless of whether it is an addition or subtraction.

While the clock rate is similar to the 1-bit scheme, only half as many clocks are required to complete the operation.

Ripple-carry Adders

The 1-bit serial adder, described above, can easily be converted into a ripple-carry parallel adder. It is simply a matter of replicating the arithmetic unit once for each bit, removing the carry/borrow flip-flops and connecting the carry/borrow outputs from one bit to the next, Figure 5. The carry/borrow input of the LSB is set to zero for no carry in an addition, and for no borrow in a subtraction.

At one CLB per bit, this design uses fewer CLBs than any other parallel adder. However, this compactness is achieved at the expense of speed; the settling time is one CLB delay per bit. By placing the CLBs of the adder adjacent to each other, interconnect delay in the ripple path can be minimized, or even eliminated.

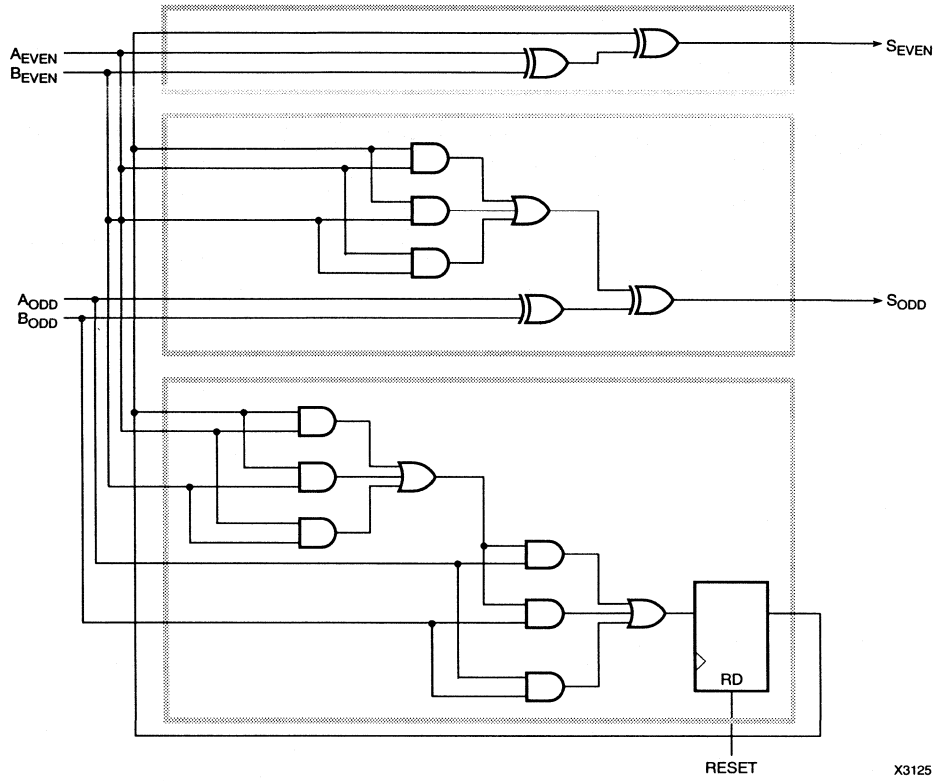


Figure 4. 2-Bit Serial Arithmetic Unit

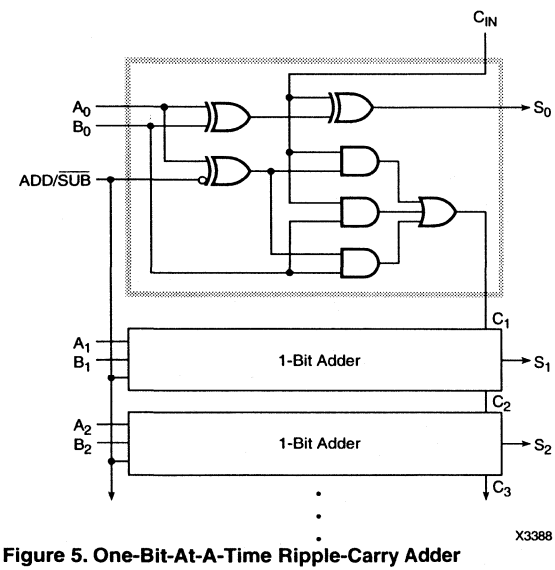


Figure 5. One-Bit-At-A-Time Ripple-Carry Adder

A faster settling time can be achieved by changing the replicated cell from a 1-bit adder to a 2-bit adder, Figure 6. The carry output and the more significant sum of each bit-pair are functions of five inputs. Consequently, each requires an entire CLB, increasing the CLB requirement to 1 1/2 per bit. However, the settling time is reduced to one CLB delay per two bits, half that of the previous design.

The 5-input function generators permit this design to be used for adders and subtracters, but not for adder/subtracters. To implement an adder/subtracter, one of the operands to an adder must be modified before being input into the adder.

For the operation A–B, there are two choices, both of which require additional XOR gates to invert one of the operands while subtracting. The technique used in the bit-serial adder and the one-bit-at-a-time adder is to invert the A-operand into the carry logic only; the A-operand is input to the sum logic unmodified. In this case, the carry/borrow input is active-high for both add and subtract, and may be tied Low if no input carry or borrow is required.

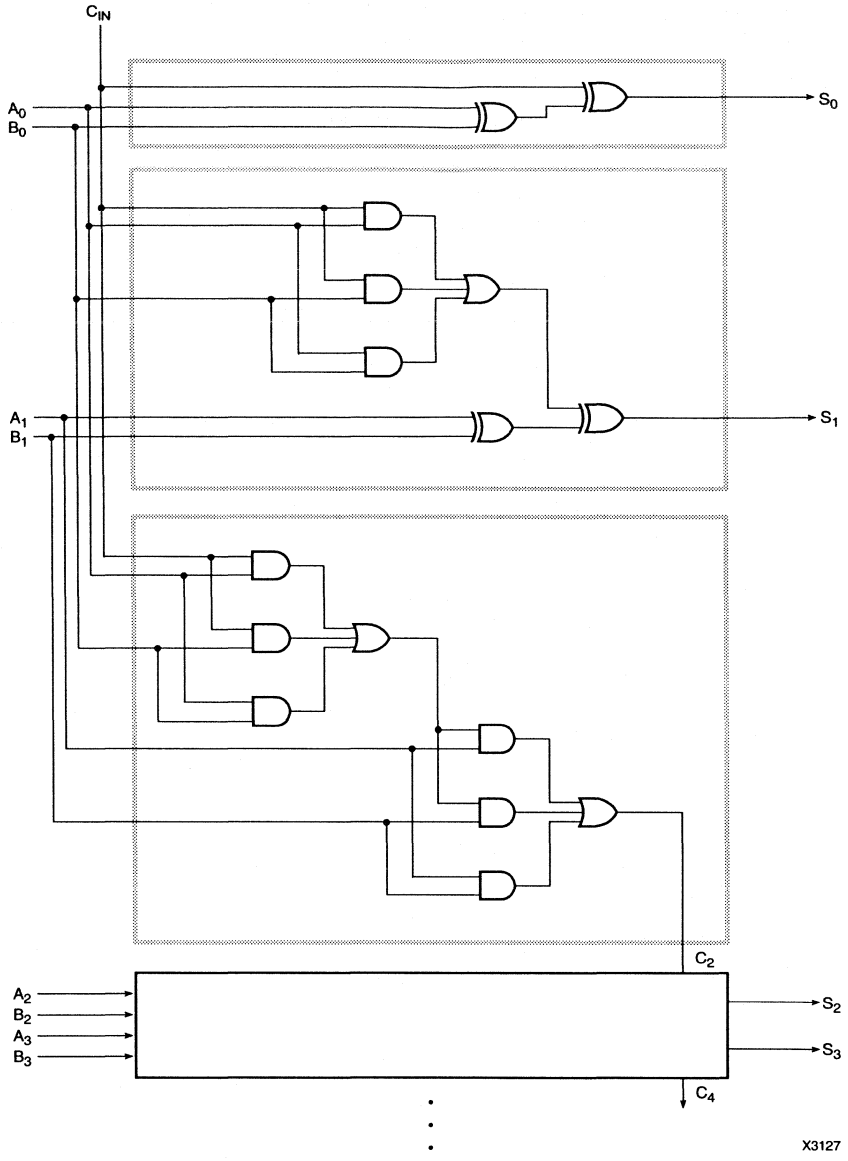


Figure 6. Two-Bits-At-A-Time Ripple-Carry Adder

A more conventional approach is to invert the B-operand into both the sum and carry logic. However, if no input borrow or carry is required, the input must be Low during an addition, and High during a subtraction.

Look-ahead-carry Adders

For faster operation in large adders, look-ahead carry look-ahead-carry technique uses two signals, Carry Generate and Carry Propagate (P and G), that are typically outputs of an arithmetic block, often of four bits. Since both of these signals do not depend on the incoming carry signal, they can be generated immediately from input data.

As the name implies, Carry Generate is asserted if the block creates an overflow (carry), regardless of incoming carry. For example, in a 4-bit adder, Carry Generate is asserted if the sum of the operand bits, excluding the incoming carry, exceeds 15.

If the block does not generate a carry by itself, but would generate a carry as a result of an incoming carry, Carry Propagate must be asserted; its assertion is optional if the block generates a carry without requiring an incoming carry. In our 4-bit example, Carry Propagate must be asserted when the sum, excluding the incoming carry, is exactly 15, and may optionally be asserted when the sum is greater.

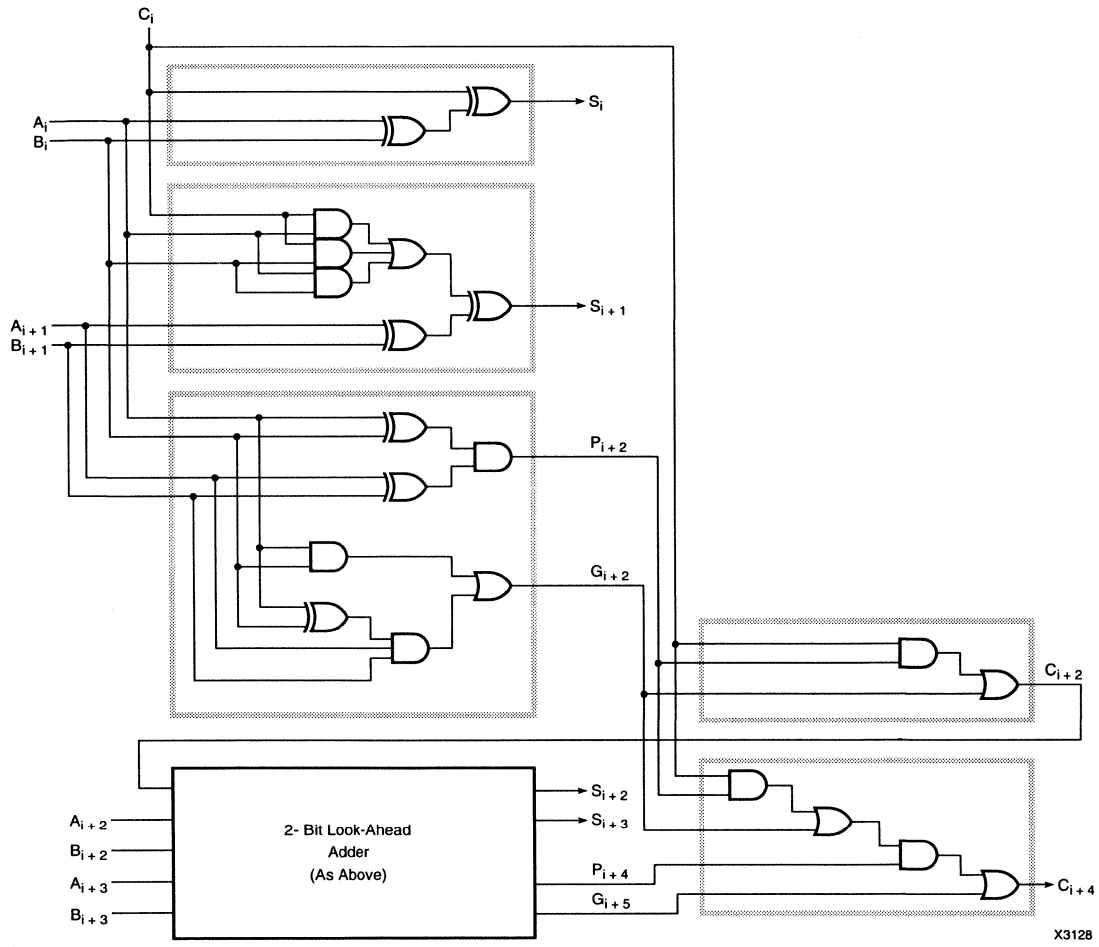


Figure 7. Four-Bits-at-a-time Adder Block with Internal Look-Ahead Carry

In XC3000 LCA devices, look-ahead carry is most effective when used to combine two 2-bit blocks into a 4-bit block that cascades using ripple carry, Figure 7. The 4-bit block has a one-CLB delay from carry in to carry out, but a two-CLB delay from carry in to the sum output of the more significant bit-pair. The delay from the operand inputs to the carry output is also two CLBs.

A 16-bit adder may be implemented in two ways. The most straightforward way is to cascade four 4-bit blocks, as shown in Figure 8(a). With this design, the carry-in-to-carry-out delay is only four CLBs, while the operand-to-sum delay is six CLBs; the operand-to-carry-out and carry-in-to-sum delays are both five CLBs The carry output is available one CLB delay before the sum, and the carry input need not be present until one CLB delay after the operands. The design requires 32 CLBs.

While a shorter carry delay may sometimes be desirable, the design in Figure 8(b) is faster overall, balancing all four delays at five CLBs. The 2-bit ripple-carry block, described in the ripple-carry section, is used to implement the most and least significant bit-pairs, and only 30 CLBs are required.

Either design can be adapted to any multiple of four bits by simply adding or subtracting 4-bit blocks in the center of the adder. The advantage over the 2-bit ripple-carry technique increases with the number of bits in the adder.

For even numbers of bits that are not multiples of four, any of the designs in Figure 9 may be used. For a 14-bit adder, the Figure 9(a) design balances all four delays at five CLBs, and requires 25 CLBs. The Figure 9(b) and 9(c) designs each use two additional CLBs, but are one CLB delay faster in the carry path. In the Figure 9(b) design the carry out appears one CLB delay before the sum, and in the Figure 9(c) design, the carry in need not be present until one CLB delay after the operand. Again, for different length adders, simply add or subtract 4-bit blocks at the center of the adder.

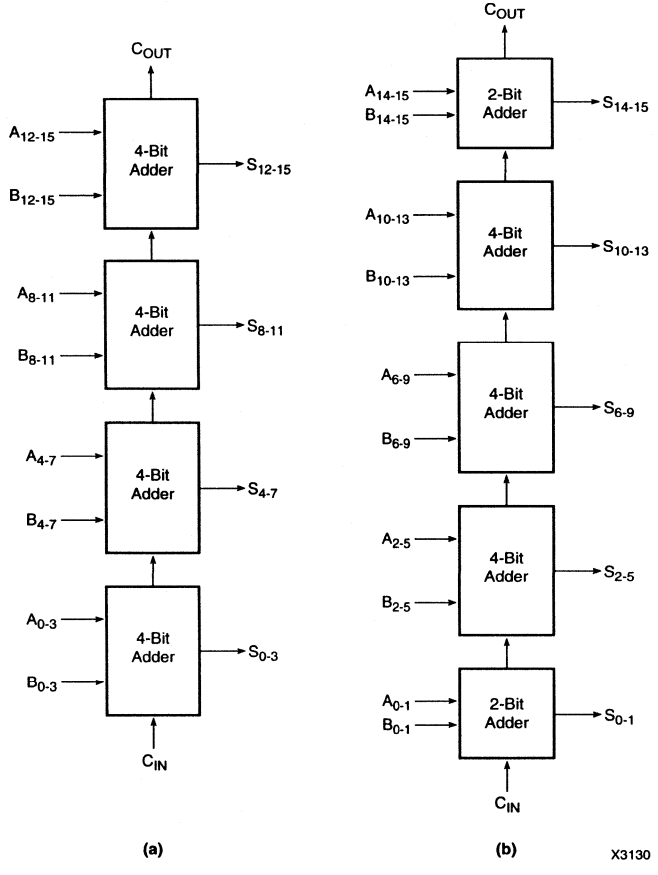


Figure 8. 16-Bit Adder Configurations

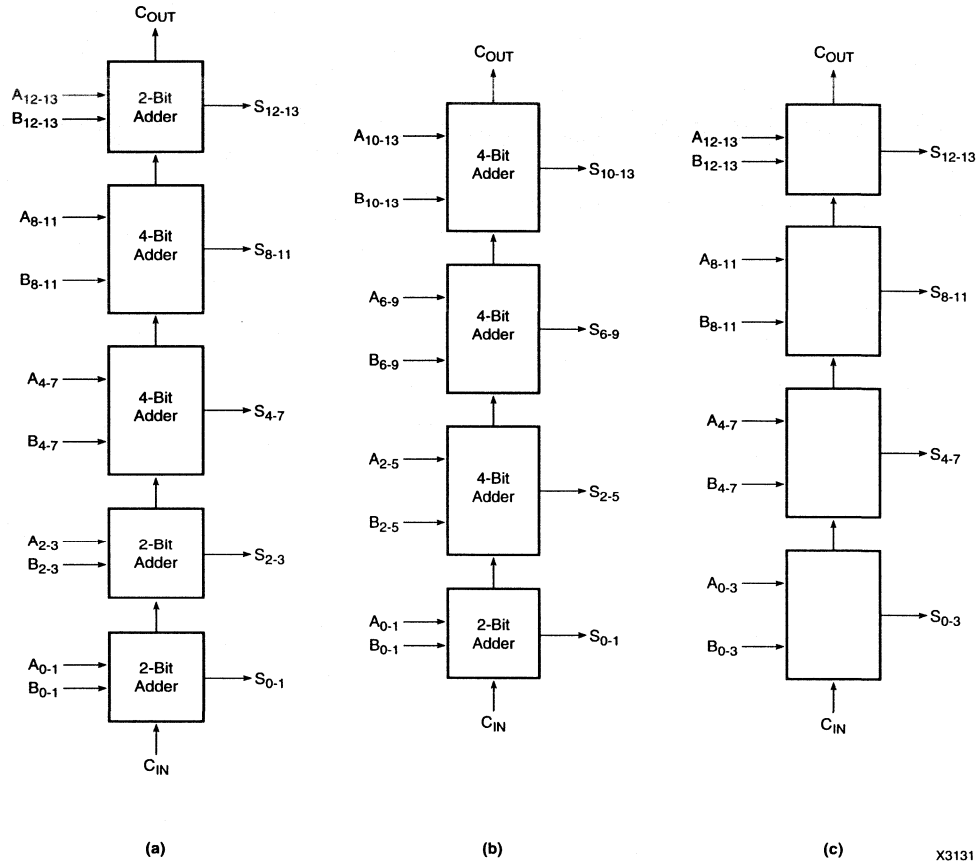


Figure 9. 14-Bit Adder Configuration

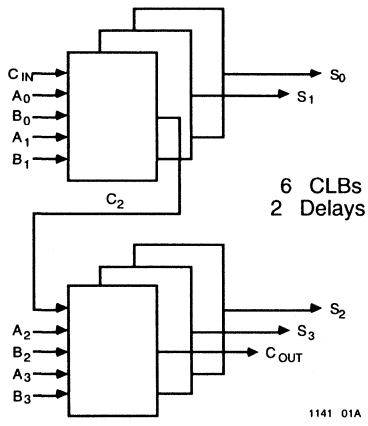


Figure 10. 4-Bit Adder

Conditional-sum Adder

Conditional-sum adders, originally described by J. Sklansky in the June 1960 issue of the IRE Transaction on Electronic Computers, reduce settling time at the expense of much higher logic complexity. The version described below was created by Matt Klein of Hewlett Packard, who modified the algorithm to fit the XC3000 architecture. With careful placement and routing, the total delay can be kept below 20 ns in an XC3100-3.

Forty-one CLBs are required, 27 of which generate one function of up to five variables, while the remaining 14 CLBs each generate two functions of four variables. Figure 10 shows how these CLBs are connected. For more information, please refer to the original paper and the Xilinx Technical Bulletin Board.

Using the Dedicated Carry Logic in XC4000

XAPP 013.001

Application Note By BERNIE NEW

Summary

This Application Note describes the operation of the XC4000 dedicated carry logic, the standard configurations provided for its use, and how these are combined into arithmetic functions and counters.

Xilinx Family

XC4000

Demonstrates

Dedicated Carry Logic

Introduction

XC4000-series CLBs contain dedicated, hard-wired carry logic to both accelerate and condense arithmetic functions such as adders and counters. Adders achieve ripple-carry delays as low as 750 ps per bit, while utilizing only half a CLB per bit. This is certainly denser than any other approach, and in most cases, faster.

As shown in Figure 1, the carry logic shares operand and control inputs with the function generators. The carry outputs connect to the function generators, where they are combined with the operands to form the sums. A conceptual diagram of a typical addition is shown in Figure 2.

Only the shared and carry inputs to the function generators are predetermined. Any function of these and the remaining inputs may be implemented. For example, in a loadable counter, the function generator may be used to both invert the counter bit, under control of the carry path, and multiplex a load value into the flip-flop. The H function generator also remains available, and the CLB flip-flops may be used in counters or accumulators.

The ripple-carry outputs are routed between CLBs on high-speed dedicated paths. As shown in Figure 3, carries may be propagated either up or down a column of CLBs. At the top and bottom of the columns where there are no CLBs above and below, the carry is propagated to the right. This enables U-shaped adders and counters to be constructed when they cannot be fitted in a single column.

The carry logic may be configured to implement add, subtract and add/subtract functions. Increment, decrement, increment/decrement and 2's-complement functions are also available.

These functions may be implemented using pre-defined CLB configurations provided in XDE. The mnemonics for these configurations, e.g., ADD-FG-CI, describe the arithmetic function supported, the CLB function generators used and the source of the carry input. While these configurations permit the dedicated carry logic to be used without detailed knowledge of its operation, the following description is provided.

Operation of the Carry Logic

A detailed and rather complex schematic of the dedicated carry logic is shown in Figure 4. Figure 5, however, is much simpler; it shows the same carry logic once it has been configured for an addition and redundant gates have been removed.

Both bits of the carry logic operate in the same way: First, the A and B inputs are compared. If they are equal, C_{OUT} is well-defined without reference to C_{IN}. When both inputs are zero, carry is not propagated and no carry is generated. Consequently, C_{OUT} must be zero. When they are both one, a carry is generated, and C_{OUT} must also be a one. In either case, C_{OUT} is equal to the A input.

If the A and B inputs are different, the carry is propagated, and C_{OUT} is equal to C_{IN}. C_{OUT} can, therefore, be created by multiplexing between the A input and C_{IN}.

This scheme is used because the multiplexers in the ripple path may be implemented using pass transistors; these introduce the least cumulative delay into this critical path.

Referring back to Figure 4, the various configuration options can now be explained. XOR-gates are provided as polarity controls for the B operands. According to a configuration bit, B may be inverted for a subtracter, or not inverted for an adder. Alternatively, the polarity may be controlled by F3 (ADD/$\overline{\text{SUBTRACT}}$) for an adder/subtracter.

The B operands may be gated out using a configuration bit in conjunction with two AND gates so that add and subtract can become increment and decrement.

To determine whether carry is propagated up or down the column of CLBs, a multiplexer selected the carry output of the CLB below or the CLB above.

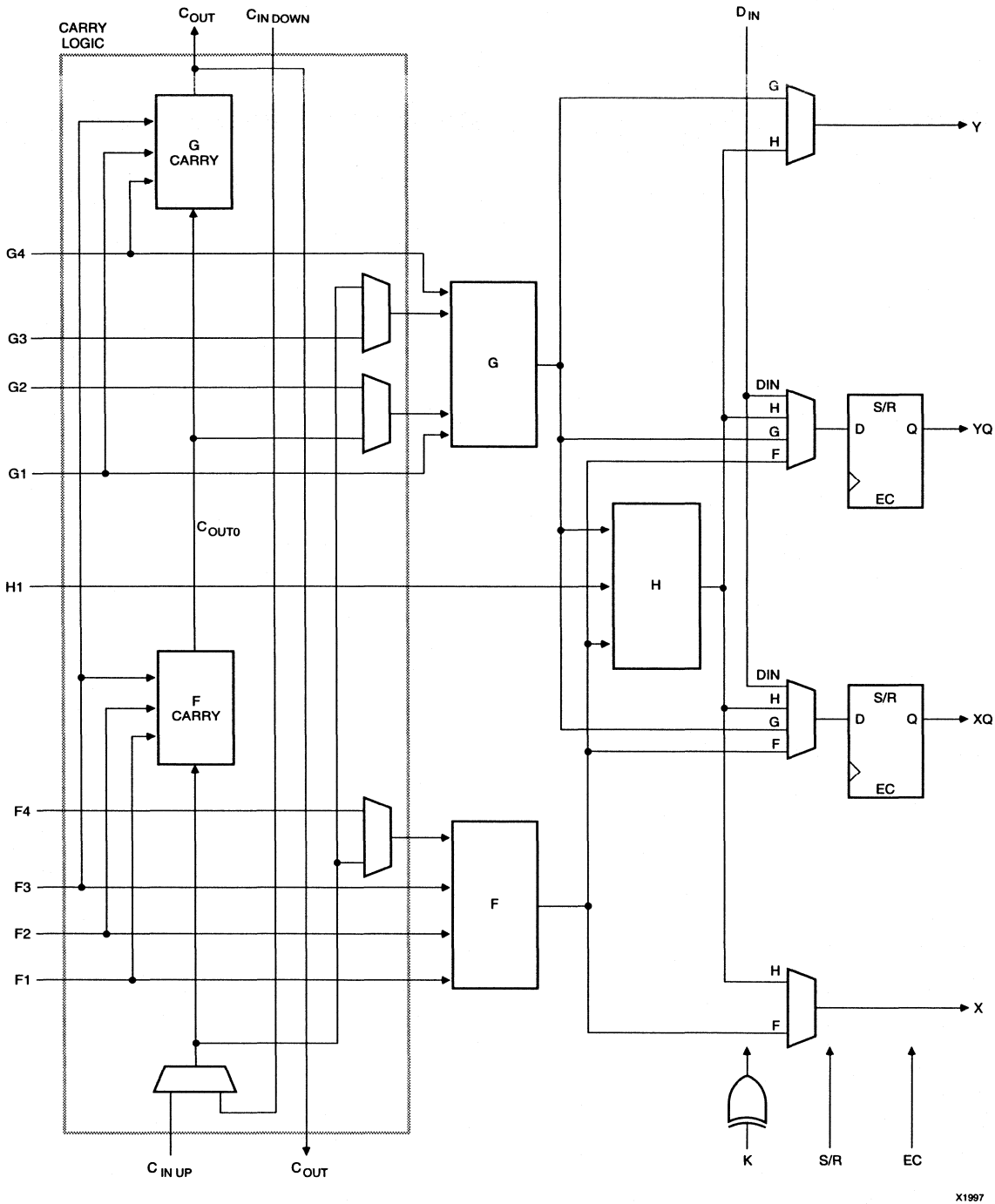


Figure 1. XC4000 Dedicated Carry Logic

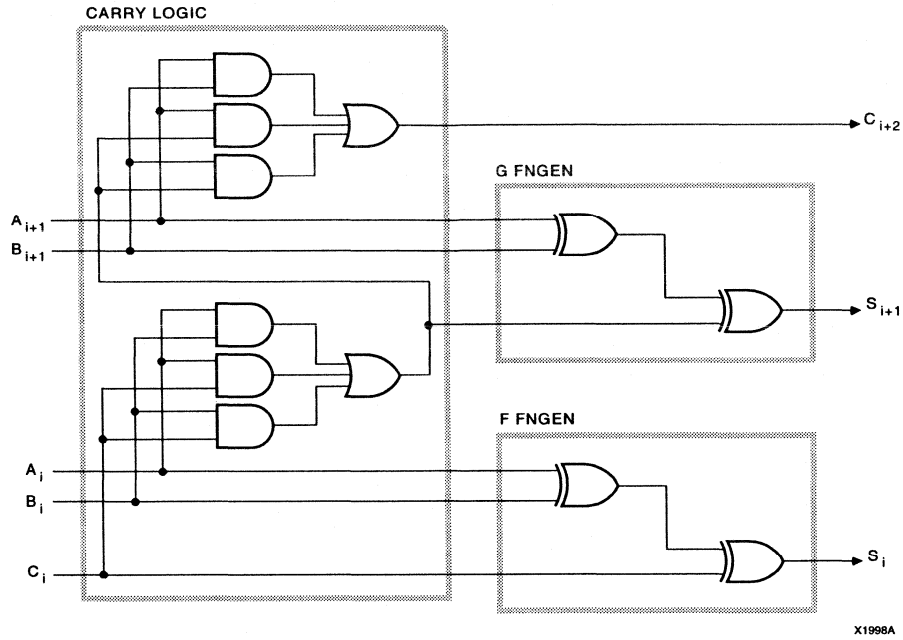


Figure 2. Conceptual Diagram of a Typical Addition (2 Bits/CLB)

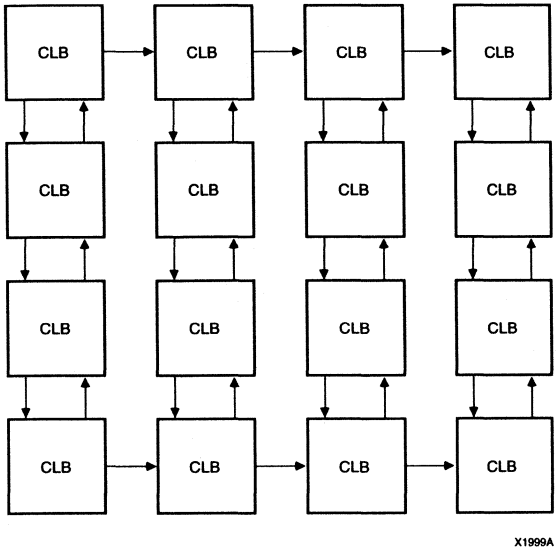


Figure 3. Carry Propagation Paths

If only one adder bit is to be implemented per CLB, the selected carry may be forced to skip the first stage of carry logic. To do this, a configuration bit is set to one and selected to replace the output of the comparator. If the bit is selected and set to zero, an initial value is forced into the carry chain.

This initial value has three sources, determined by the configuration bits. The first source is the configuration bit used to gate out the B operand. When this bit is a one, a 2-operand function is performed, and a one at the carry input provides add-with-carry or subtract-without-borrow (borrow is active Low). When the bit is a zero, a 1-operand function is performed, and the carry chain is initialized with a zero.

The second source is $\overline{F3}$. If $\overline{F3}$ is not selected as the add/subtract control, it is a free input to the carry chain. If it is used to control addition and subtraction, it provides a zero or one such that the initial carry/borrow is unasserted in both cases.

The final source is F1. When initialization is selected, this is a free input to the carry chain.

The second stage of the carry logic may also be skipped, in the same way as the first stage. However, there is no initialization function in the second stage.

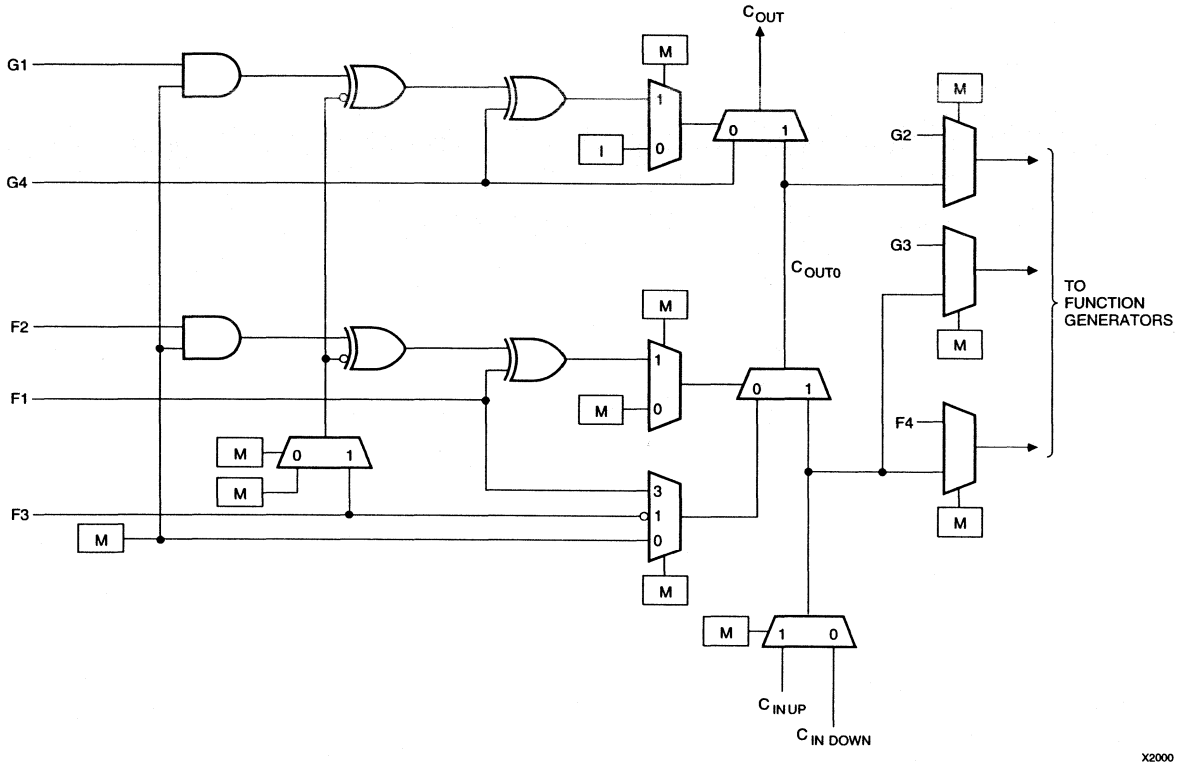


Figure 4. Detail of Dedicated Carry Logic

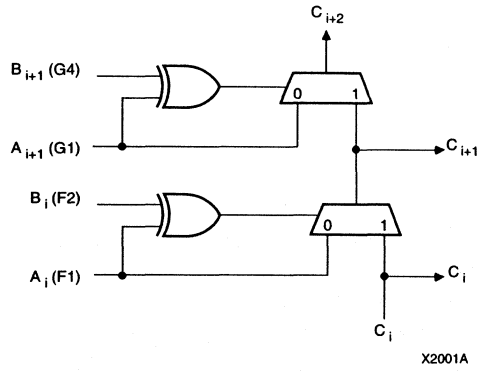


Figure 5a. Effective Carry Logic for a Typical Addition

A	B	C	C_{OUT}	
0	0	0	0	$A = B, C_{OUT} = A$
0	0	1	0	
0	1	0	0	$A \neq B, C_{OUT} = C_{IN}$
0	1	1	1	
1	0	0	0	
1	0	1	1	
1	1	0	1	$A = B, C_{OUT} = A$
1	1	1	1	

X2002A

Figure 5b. Effective Carry Logic for a Typical Addition

2-Operand Functions

Adders

An adder implemented with the dedicated carry logic must have at least two sections: a main section and an initialization section. In the main section, shown in Figure 6, one or two bits of the adder are implemented in each CLB, and C_{IN} is taken from the dedicated interconnect. Three standard CLB configurations are provided for this purpose: ADD-FG-CI is a two-bit adder, while ADD-F-CI and ADD-G-CI are one-bit adders with the add occurring in F or G, respectively.

C_{IN} can only be driven by other carry logic. At the least significant end of the adder, special attention must paid to ensure that the carry path is initialized correctly. This is the function of the initialization section.

The design of the carry logic does not provide for the implementation of two adder bits in the initializing CLB. However, a CLB may be used to initialize the carry path and implement the LSB of the adder. The standard CLB configurations for this are ADD-G-F1 and ADD-G-F3-. In both cases, the addition occurs in G, and the carry input is F1 or F3, respectively.

The use of this technique may create bussing difficulties if other parts of the LCA device have the two LSBs implemented in the same CLB. A second approach that avoids this problem uses a CLB to initialize the carry path without implementing part of the adder.

Four standard CLB configurations are provided for this purpose: FORCE-F1 and FORCE-F3- allow F1 and $\overline{F3}$, respectively, to be used as the carry input, while FORCE-0 and FORCE-1 initialize the carry path with a fixed zero or one, respectively. FORCE-0 and FORCE-1 only involve the carry logic, and all the non-carry resources of the CLB are available for other uses.

Optionally, the adder may have a third section at the most significant end, used to create a carry output (other than on the dedicated interconnect) or to detect overflow. Two situations must be considered: where the most significant CLB contains two bits of the adder, and where it contains only one.

If it contains only one bit of the adder, the standard CLB configuration, ADD-F-CI, in Figure 7 should be used. Both C_{IN} and the most significant carry are available as inputs to the G function generator. The most significant carry may be passed through this, or XOR-ed with C_{IN} to detect twos-complement overflow.

Where both carry and overflow are required, overflow should be generated in the same CLB as the most significant bit. The most significant carry is passed to C_{OUT}, and an additional CLB may be configured to route it to either the F or G output. The EXAMINE-CI configuration is provided for this purpose.

If the most significant CLB contains two bits of the adder, the situation is more complex. As shown in Figure 8, the ADD-F-CI configuration should again be used, despite the need for a 2-bit adder. The most significant bits of the operands should be connected to the G1 and G4 inputs, C_{OUT0} selected as the G2 input, and the G function generator manually programmed as if the configuration were ADD-FG-CI. This causes the most significant sum to be generated at the G output. However, the second stage of carry logic will be bypassed.

An additional CLB can then be used to generate the carry and the overflow. This should be configured as ADD-F-CI and the most significant bits of the operands connected to F1 and F2 in addition to the previous connection. This causes the carry stage, bypassed in the previous CLB, to be implemented in the first stage of this additional CLB. In this way, the necessary carries are available in the G function generator for overflow detection as described above.

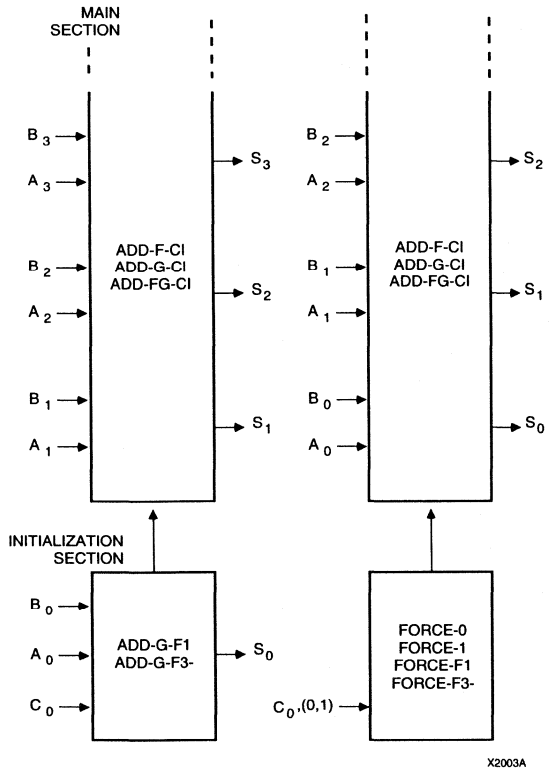


Figure 6. Main and Initialization Sections of Adder

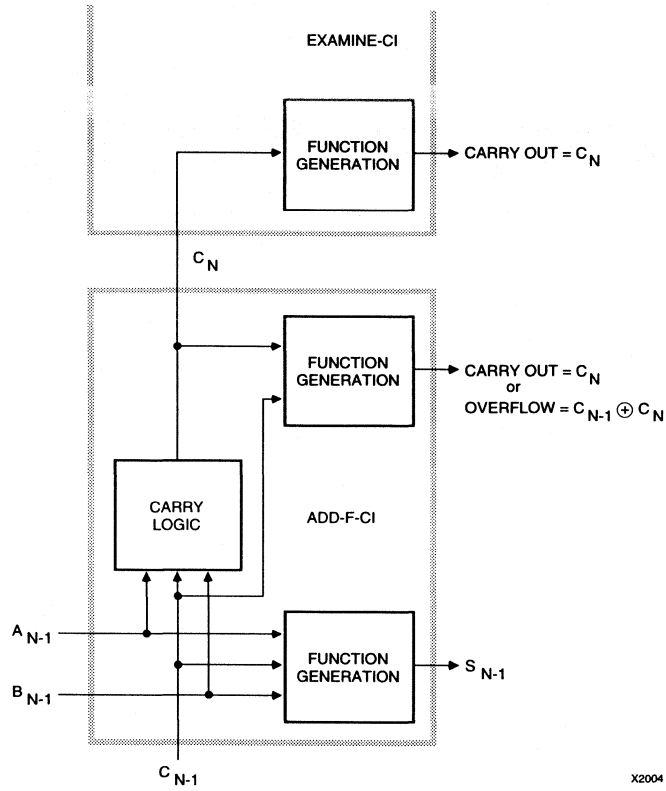


Figure 7. Carry-Out and Overflow Generation

The F function generator may be manually programmed to create the most significant carry from the operand bits and C_{IN}. This is permissible as the operation of the carry logic is independent of the function generators.

Subtracters

Subtraction is, in most respects, identical to addition. The subtraction may be written in terms of an addition as follows:

$$A - B = A + (-B)$$

Multiplication by -1, or two's complementing, is performed by logically inverting the operand and adding one. The final form of the subtraction becomes:

$$A - B = A + \overline{B} + 1$$

Using CLB configurations with a SUB prefix, in place of ADD, causes the B operand to be inverted both into the carry logic and within the function generator. The one can be added by forcing the carry into the adder to be High.

An alternative interpretation is that the inversion changes the adder into a subtracter, with the carry becoming an active-Low borrow. Consistent with the first interpretation, the carry input must be High for borrow not to be asserted. If the carry input is Low, the operation is A - B -1.

Apart from using CLB configurations with the SUB prefix and ensuring that carry-in has the right polarity, subtracters may be constructed in the same way as adders. Equivalent configurations exist for all three sections of the subtracter. The only point to remember is that, when manually configuring function generators for the most significant output or carry output, the B operand must be inverted. The definition of overflow does not change.

One configuration that exists for subtraction, but not for addition, is SUB-G-1. In this configuration, the least significant bit of the subtraction takes place in G with the carry input internally forced to a one (no borrow).

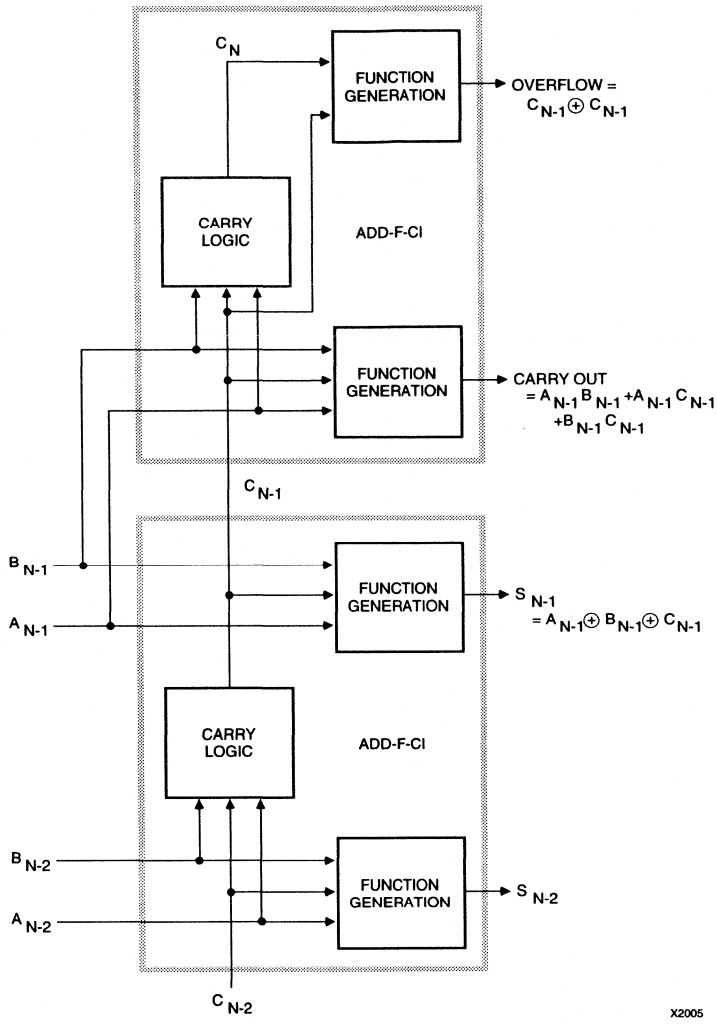


Figure 8. Carry-Out and Overflow Generation with Duplicated MSB

Adder/Subtracters

The adder may be converted to an adder/subtracter by making the inversion of the B operand programmable. This is accomplished using CLB configurations with an ADDSUB prefix.

The ADD/$\overline{\text{SUBTRACT}}$ control is connected to F3, and controls the operation of both the carry logic and the F function generator. If the configuration uses the G function generator, ADD/$\overline{\text{SUBTRACT}}$ must also be connected to G3.

The carry input to the adder/subtracter must be determined by the operation being performed. When an add is in progress, it must be Low for a carry not to be asserted, and it must be High for a borrow not to be asserted during a subtraction.

This will generally preclude the use of FORCE-0 and FORCE-1 to initialize the carry chain. Otherwise, the adder/subtracter is constructed in the same way as the adder, but using CLB configurations with the ADDSUB prefix.

As in the subtracter, the programmable operand inversion must be remembered in any function generators that are manually configured

1-Operand Functions

Incrementers

Essentially, an incrementer is an adder with one operand zero, and the carry input asserted. Consequently, incrementers are constructed in the same way as adders, but using CLB configurations with an INC prefix. These gate out the B operand.

The carry input should be High to increment the A operand, and Low to pass it unchanged. Alternatively, it may be fixed High for permanent incrementation. This may be accomplished using CLB configurations equivalent to those used to initialize adders. In addition, INC-G-1 and INC-FG-1 allow the carry chain to be initialized with the carry asserted, along with one or two bits of the function.

Decrementers

These are subtracters with the B operand zero and a borrow asserted. CLB configurations with a DEC prefix gate out the B operand before it is inverted. The carry input should be Low to decrement the A operand, and High to pass it.

Alternatively, a fixed Low may be used. DEC-G-0 and DEC-FG-0 provide this, along with one or two bits of the function. FORCE-0 may also be used.

Incrementer/Decrementers

Not surprisingly, these are constructed in the same way as adder/subtracters, but using cells with an INCDEC prefix that gate out the B operand. When increment is selected, the carry input should be High to increment or Low to pass. When decrement is selected, the carry should be Low to decrement or High to pass. INCDEC-FG-0 implements two least significant bits of the incrementer/decrementer with the carry or borrow input permanently asserted.

2's Complementers

The traditional two's-complement procedure, invert-and-add-one, is not appropriate for use with the dedicated carry logic. In the increment configuration, the A operand cannot be inverted at the input to the carry logic, and using a subtracter for 0 - B consumes unnecessary resources routing the zero operand.

The answer it to replace invert-and-increment with decrement-and-invert, which produces the same result. A conventional decrementer is constructed, and an additional output inversion is programmed into the function generators.

The use of a function generator input allows this inversion to become programmable. In conjunction with control of the carry input, this programmable inversion may be used to twos complement a number or pass it, as required.

Counters

Up Counters

An up counter is constructed by combining an incrementer with a register, as shown in Figure 9. Typically, the register in the same CLBs as the incrementer is used, and the sum outputs should be routed to this register. The output of the register is fed back as the input to the incrementer. Each clock, the register is loaded with a value one greater than its previous value.

Any incrementer may be used. If it has the ability to increment or pass the operand, this feature may be used as a count enable.

As shown in Figure 10, counters may easily be made loadable by adding a multiplexer into the function generators. This multiplexer selects between the incrementer output and the value to be loaded as the source for the register.

Down Counters

Down counters are constructed in the same way as up counters, but using decrementers in place of incrementers.

Up/Down Counters

Incrementer/decrementers are used for up/down counters. The only significant difference comes in the loadable counter. Because the INC/$\overline{\text{DEC}}$ control is an

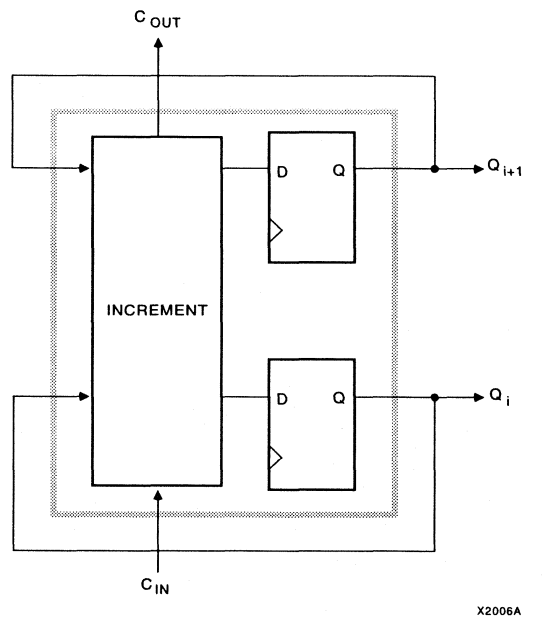


Figure 9. Typical Counter CLB

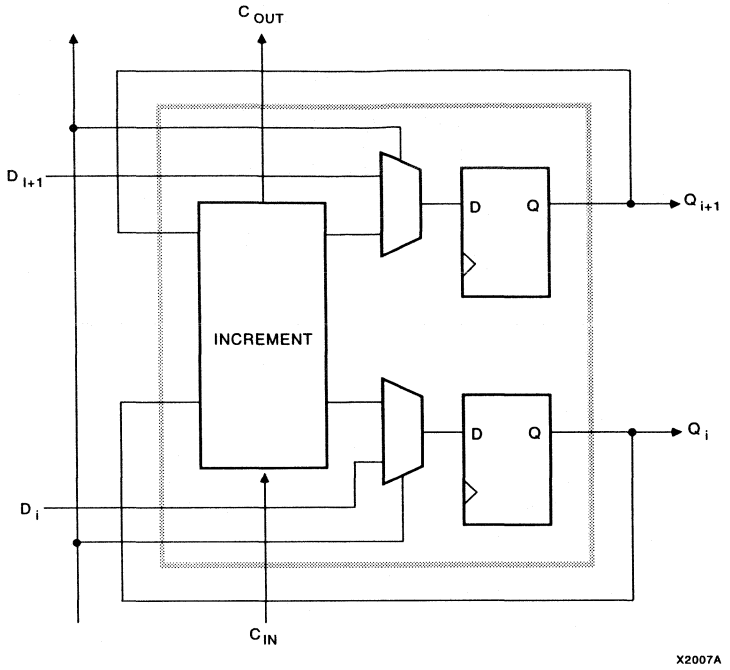


Figure 10. Typical Loadable Counter CLB

input to the function generators, there are not enough inputs available for the load function. One CLB must be used for each bit of the counter, and there are several ways in which this can be organized.

One possibility is to use a CLB configuration that only implements one bit of the incrementer/decrementer function, as shown in Figure 11. The H function generator can then be used as the load multiplexer. The H1 input acts as the Parallel Enable, and the value to be loaded is passed through the second function generator.

A better choice is to construct the incrementer and decrementer separately in two columns of CLBs with two bits per CLB, as shown in Figure 12. The decrementer is connected as a conventional loadable down counter. In the incrementer, the function generators are modified with a multiplexer, as is it were to be a loadable up-counter. However, the register is not connected, and data is not fed back.

Instead, the input to the incrementer is taken from the output of the down counter, and the incrementer output is routed to what would have been the down-counter load input. The value to be loaded is input to the multiplexer attached to the incrementer.

The load control of the down counter becomes the up/down control, selecting the output of either the incrementer or the decrementer. Data is loaded by replacing the incrementer output with the value to be loaded, and selecting count up. An external gate may be required to force the up/down control.

This second approach has the advantage that its layout is compatible with other functions that implement two bits per CLB. More importantly, however, it is faster. The incremental carry delay is incurred per CLB, not per bit, and implementing two bits per CLB halves the number of carry delays. Also, the set-up time on the up/down control is much shorter. The up/down control need only select the output of the incrementer or decrementer, instead of selecting the increment or decrement function before carry/borrow propagation can begin. Both the incrementer and decrementer operate in parallel, starting immediately after the clock.

Alternatively, an incrementer/decrementer may be implemented in one column of CLBs, with the register and load multiplexers implemented in a second column. A count-enable multiplexer can be built into the same function generator as the load multiplexer. If this is placed logically in front of the load multiplexer, the load control takes precedence over the Count Enable.

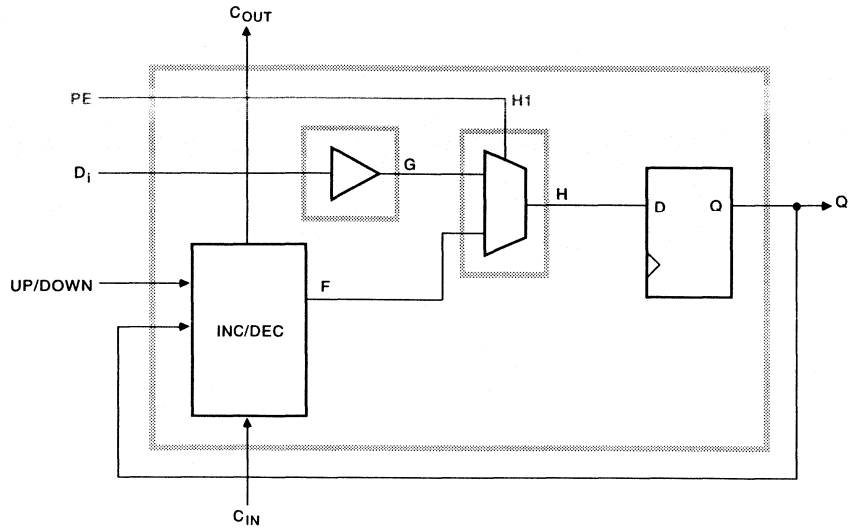


Figure 11. Typical Up/Down Counter CLB

X2008A

This scheme eliminates the additional gating required to ensure that the counter is enabled and counting up during a load. The Load and Count Enable controls are both fast, but the set-up time for the up/down control is similar to the carry-propagation delay.

Timing Analysis

Typically, the critical delay is from the carry input or operand LSB to the output MSB, carry output or overflow flag. As shown in Figure 13, this delay has three parts: The delay onto the carry chain from the input, the delay from the carry chain to the output and the delay of the intervening CLBs.

If part of the function is performed in the CLB that initializes the carry chain, the delay onto the chain is the greater of the operand-input-to-C_{OUT} (T_{OPCY}) and the initialization-input-to-C_{OUT} (T_{INCY}) delays. If a CLB is used for initialization only, separate delays must be calculated from the least significant operand input and the initialization input, taking into account the different number of intervening CLBs.

The output delay (T_{SUM}) is from C_{IN} to the output. Each intervening CLB introduces a T_{BYP} delay.

To calculate the minimum clock period in a counter, the clock-to-output delay and a routing delay must be added to the operand input delay. Typically in a -5 part, this routing delay is 1.5 ns; but this must be verified by simulation after the implementation is complete. The output delay must be replaced with the equivalent set-up time, and the intervening CLBs taken into account, as in the basic delay calculation.

Configuring the Carry Logic

The dedicated carry logic is accessed through the use of hard macros. These are blocks of CLBs that are configured and routed in the XACT Design Editor (XDE), and then converted to macros using HMGEN. When the symbol for the macro is used in a schematic, the relative placement and configuration of the CLBs are retained.

Individual CLBs are configured using the EditBlk command. Within the Block Editor, the ConfigCarry command provides a list of the standard CLB carry configurations. Once a selection is made, the mnemonic for the configuration appears in the Block Editor screen.

The selection causes the F4, G2 and G3 tags to be set according to the chosen configuration, and the appropriate functions are entered into the F and G function generators. If the tags or function generators had been previously defined, they are not overwritten. If the settings values are required, any previous settings must be cleared before selecting the CLB configuration.

The direction of the carry propagation, up or down, must be selected by the C_{DIR} tag. In addition, check that the carry inputs and outputs are routed appropriately by the C_{IN} and C_{OUT} tags.

If the standard configuration needs to be modified, the changes are simply entered on the Block Editor screen. Once editing of the block is complete, a carry route must be added between adjacent CLBs.

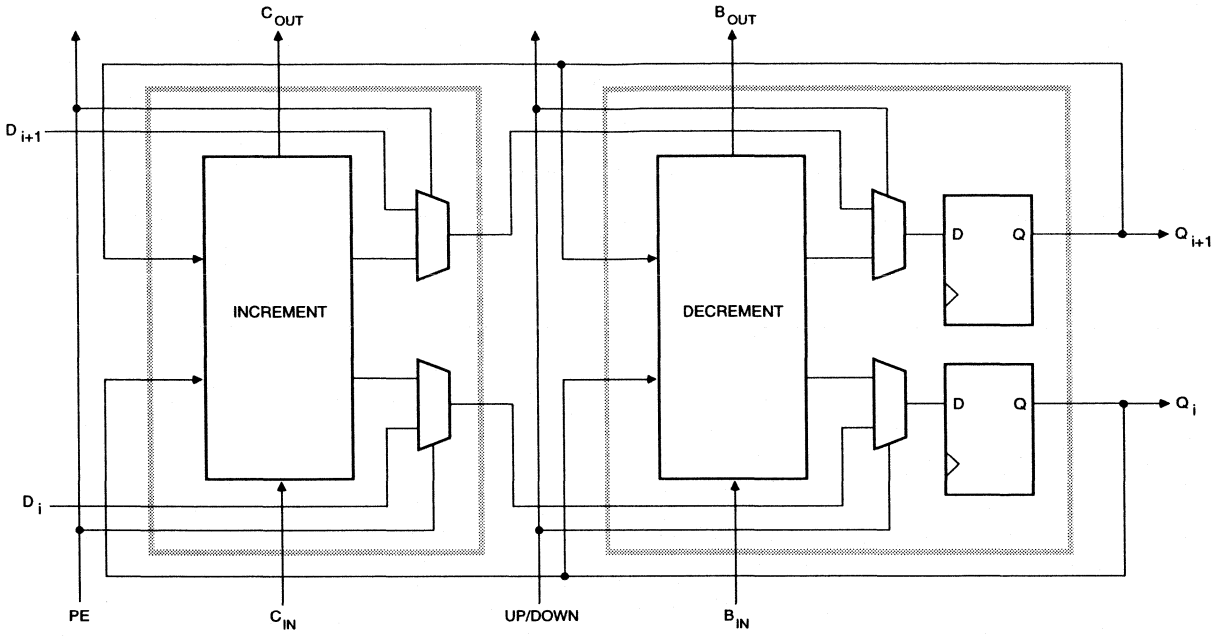


Figure 12. Up/Down Counter with Separate Incrementer and Decrementer

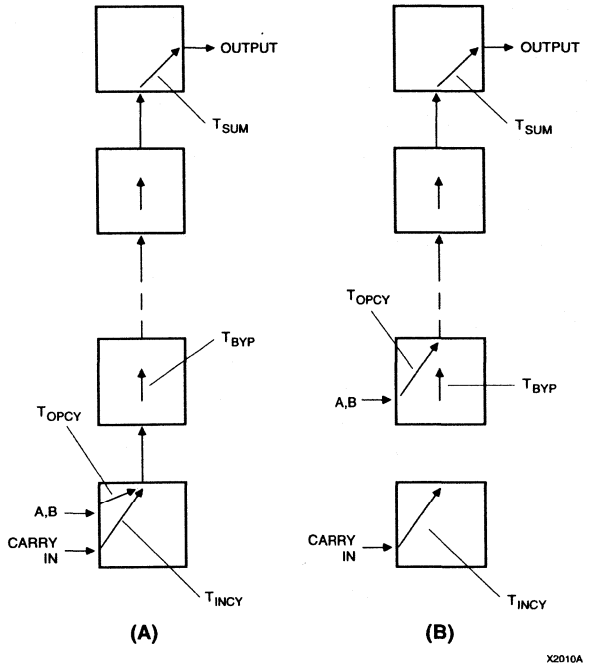


Figure 13. Carry-Logic Delay Paths

Estimating the Performance of XC4000 Adders and Counters

XAPP 018.000 Application Note By BERNIE NEW

Summary

Using the XC4000 dedicated carry logic, the performance of adders and counters can easily be predicted. This Application Note provides formulae for estimating the performance of such adders and counters.

Xilinx Family

XC4000

Demonstrates

Dedicated Carry Logic

Introduction

In most LCA designs, performance cannot be estimated with any accuracy until after implementation. This is because the performance is affected by routing delays; and, prior to implementation, these are not known. However, in adders and counters using the XC4000 dedicated carry logic, delay estimation is possible.

The carry path in an adder uses dedicated interconnects between CLBs. These interconnects introduce a fixed delay, even when the carry passes from one CLB column to the next at the top or bottom of the array. This permits the routing delay to be incorporated into the CLB specifications published in the data book. Consequently, the propagation delay through an adder can be calculated directly from the data book specifications.

For a typical adder, this calculation can be reduced to a simple formula. In an XC4000-5*, the maximum propagation delay from the operand input to the sum output of an N-bit adder is approximately

$$t_{pd} = 8.5 + 0.75N \text{ ns}$$

This estimate does not include the delay from the operand source register to the adder or any additional delay reaching the destination register. However, it is still a useful benchmark.

This formula applies only to simple ripple-carry adders. However, such adders are adequate in most situations; conditional-sum and other adder-acceleration schemes are only appropriate for adders longer than 24 bits.

For an N-bit counter, the minimum clock period that permits the carry path time to settle is approximately

$$t_{clk\text{-}clk} = 13 + 0.75N \text{ ns}$$

The following discussion describes how these formulae were derived, under what conditions they apply, and the corrections that must be made when these conditions are not met.

*Based on the December 1991 Data Sheet

It must be stressed that these formulae are intended only as initial estimation tools. They do not replace the full timing analysis that should be performed after implementation.

Adders

The above formula for an N-bit adder assumes that N is even and that the adder (excluding any carry-chain-initialization logic) is implemented in N/2 CLBs, Figure 1. In this organization, the least significant two bits share a CLB, and the delay onto the carry chain in this CLB is T_{OPCY}. The most significant two bits also share a CLB, and, in this CLB, the delay from the carry chain to the most significant output is T_{SUM}. The intervening N-4 bits contribute a T_{BYP} delay for every two bits. Because the carry signal uses dedicated interconnects, there effectively is no routing delay in this path.

This permits the propagation delay to be expressed as follows.

$$t_{pd} = T_{OPCY} + (N\text{-}4)/2 \times T_{BYP} + T_{SUM}$$

For an XC4000-5

$$t_{pd} = 5.5 + (N\text{-}4)/2 \times 1.5 + 6 \text{ ns}$$
$$= 8.5 + 0.75N \text{ ns}$$

In adders with this organization, part of an additional CLB must be used to initialize the carry chain; and this CLB may be used to create a carry input. Delays from this carry input may also be estimated using the above formula. The T_{OPCY} delay onto the carry chain is replaced by a T_{INCY} delay onto the carry chain plus an additional T_{BYP} delay. Conveniently, these delays are equal; delays from the carry input and from the LSB of the operand are the same.

If a carry output or overflow flag is generated, an additional CLB at the most significant end of the counter is required. Consequently, the delay to these outputs is one T_{BYP} delay (1.5 ns) longer than to the MSB output.

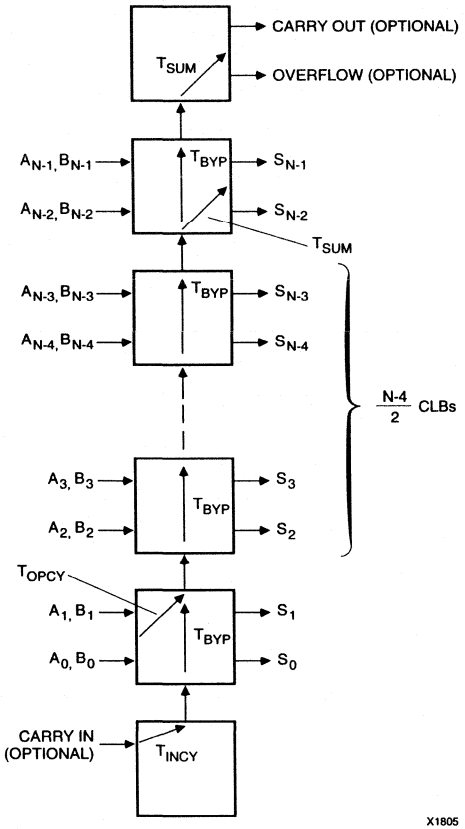


Figure 1. Basic Adder Organization

An alternative organization for the adder (Figure 2) places the LSB and the MSB into individual CLBs, with each pair of intervening bits sharing a CLB. This organization results in one additional pair of intervening bits. Consequently, an additional T_{BYP} delay (1.5 ns) is incurred in all paths using the carry chain.

In this organization, the carry chain can be initiated in the CLB used to implement the LSB of the adder. In this case, the delay from the carry input is faster that the delay from the operand LSB. The delay is reduced by T_{OPCY} minus T_{INCY}; again, this is 1.5 ns.

In the CLB implementing the MSB of the adder, it is possible to generate either a carry output or an overflow flag, but not both. The delay to this additional output is the same as to the MSB of the adder. If both carry and overflow are required, an additional CLB must be used for one of them, and the signal generated in this CLB incurs an additional T_{BYP} delay (1.5 ns).

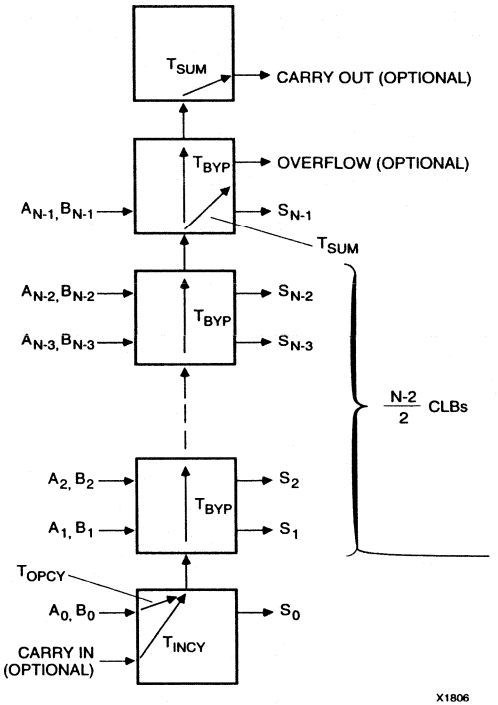


Figure 2. Alternative Adder Organization

The organization of an adder with an odd number of bits is a hybrid of the two organizations discussed above. One end of the adder has two bits sharing a CLB, while the other end has a single bit in a CLB. Either end may have the shared CLB, and this end matches the first organization. The other end, with a single bit in a CLB, matches the second organization.

For delay calculations, the number of bits should be rounded up to an even number. The basic delay formula can then be applied without correction.

If the single bit is at the most significant end of the counter, the least significant end of the counter matches the first organization. If a carry input is provided, the delay from this input must use the adjustment for the first organization. The most significant end of the counter matches the second organization, and delays to carry-out or overflow must use the corrections for that organization. If the single bit is at the least significant end of the counter, this situation must be reversed.

The set-up time from the carry chain to flip-flops in the same CLB matches the CLB output delay from the carry chain. Consequently, all the delays discussed above can

also be considered as set-up times to the register contained in the same CLBs as the adder. Different delay formulae must be derived for adders not organized with two bits per CLB.

Subtracters and Adder/Subtracters

The performance analysis, described above, also applies to subtracters and adder/subtracters. In an adder/subtracter, however, there is an additional add/subtract control input that must be considered.

To estimate the add/subtract-to-carry delay, the operand-to-output delay, appropriate to the organization, must be modified. Its operand-to-carry delay (T_{OPCY}) must be replaced by an add/subtract-to-carry delay (T_{ASCY}). This causes an increase of 0.5 ns.

This increase also applies to delays from the add/subtract input to the carry output or overflow flag.

Counters

The performance of carry-logic-based counters implemented with two bits per CLB can be estimated in a similar way. These include loadable up counters and down counters, and non-loadable up/down counters.

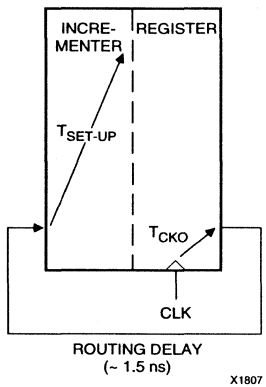


Figure 3. Basic Counter Configuration

As stated above, all of the delay estimates may also be considered set-up time estimates when using the register in the same CLB as the adder. This also applies to an incrementer or decrementer used to implement a counter.

To estimate the minimum clock period, the delay from the register to the incrementer/decrementer must be added to the incrementer/decrementer set-up time, as shown in Figure 3. This additional delay involves a clock-to-output delay (T_{CKO} = 3 ns) plus a typical routing delay of 1.5 ns. Consequently, the minimum clock period for an N-bit counter is

$$t_{clk\text{-}clk} = 13 + 0.75N \text{ ns}$$

This assumes a counter with an even number of bits, organized in the same way as the first adder. If the alternative organization is used, the clock period must receive the same 1.5 ns adjustment that was applied to the adder delay. The carry input to the incrementer/decrementer may be used as a count enable, and the same set-up-time estimate applies. Also, the carry output may be used as terminal count. The delay from the clock to the terminal count output is the minimum clock period with any correction that might be necessary for estimating the carry-out delay with the equivalent organization.

In a non-loadable up/down counter, the add/subtract control becomes up/down. The estimate for add/subtract-to-output delay is equivalent to the set-up time for the up/down control. Loadable up/down counters cannot be organized such that these formulae can be applied.

Other Speed Grades

Similar estimation formulae can be derived for other speed grades. For an XC4000-6, the basic operand-to-output delay for an N-bit adder is

$$t_{pd} = 11 + N \text{ ns}$$

The 1.5 ns correction factor, used above, increases to 2 ns, in all cases. The delay increase from the add/subtract input becomes 1 ns.

The minimum clock period for a counter is

$$t_{clk\text{-}clk} = 18 + N \text{ ns}$$

Calculating XC7000 Arithmetic Performance

XAPP 032.001

Application Note By JEFFREY GOLDBERG

Summary

This Application Note describes how to estimate the performance of arithmetic circuits that are implemented using the XC7000 dedicated carry circuitry.

Xilinx Family

XC7200, XC7300

Demonstrates

Dedicated Carry Logic

Introduction

Xilinx XC7000-family EPLDs contain dedicated fast arithmetic carry nets running directly between adjacent Macrocells and Function Blocks.This carry logic supports fast adders, subtracters, accumulators, and magnitude comparators. This carry logic is enhanced in XC7300 devices by the addition of FB carry look-ahead circuits. The use of data-sheet timing parameters to calculate the performance of wide arithmetic functions is explained below.

Performance Calculation – XC7200

Performance calculations are based on the circuit shown in Figure 1, which adds two n-bit wide numbers and stores the sum in an output register; input data comes from two on-chip registers. The carry propagation path inside the adder determines the maximum operating frequency of this circuit. The data sheet defines three carry propagation delays.

- t_{PDT1} is the carry delay through one Macrocell, i.e., from the output of one Macrocell to the output of the adjacent Macrocell in the same Function Block.
- t_{PDT8} is the carry delay through eight Macrocells, i.e. from the output of the first Macrocell in a Function Block to the output of the ninth Macrocell in the same Function Block. This specification is less than eight times t_{PDT1} because of test-guardbanding.
- t_{PDT9} is the carry delay through a whole Function Block plus the delay between Function Blocks, i.e. the delay from the output of any particular Macrocell in one Function Block to the output of the equivalent Macrocell in the adjacent Function Block. In some devices, the additional delay when crossing a Function Block boundary makes t_{PDT9} larger than the sum of t_{PDT1} and t_{PDT8}

An 18-bit adder is used to illustrate the performance calculation. First, draw a block diagram showing how the adder is mapped into the Function Blocks, as shown in Figure 2. Place the least significant bit of the adder into the least significant Macrocell of the function block and place the remaining 8 bits into the remaining Macrocells, then fill the second function block with the nine most significant bits.

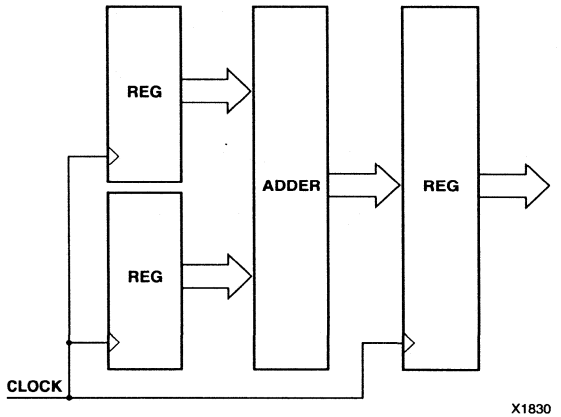


Figure 1. Arithmetic Performance Benchmark Circuit

The carry propagation delay is the sum of three ingredients.

- the time to generate the sum for the LSB, $1/f_{CYC}$.
- the delay through the first Function Block, t_{PDT9}.
- the delay inside the second Function Block, t_{PDT8}.

Using XC7236A-16 values from the data sheet makes this a total carry delay of 23 ns.

If the adder is made one bit wider, it crosses one additional Function Block boundary. Consequently, the total delay increases by the difference between t_{PDT8} and t_{PDT9}; the 19-bit adder settles in 24 ns.

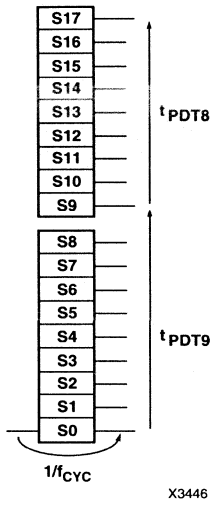


Figure 2. Adder Block Diagram

Performance Calculation – XC7300

Performance calculations are also based on the circuit in Figure 1. In the XC7300 architecture, each Function Block provides a carry lookahead generator capable of anticipating the carry across all nine Macrocells. The carry lookahead generator reduces the ripple-carry delay of the wide arithmetic function to that of the first nine bits plus the lookahead carry delay of the higher order Function Blocks. The XC7300 timing model defines two carry delays.

- t_{CARY8} is the carry delay through eight Macrocells, i.e., from the output of the first Macrocell in a Function Block the output of the ninth Macrocell in the same Function Block.
- t_{CARYFB} is the carry lookahead delay per additional Function Block

The carry propagation delay is the sum of three ingredients.

- the time to generate the sum for the LSB, $1/f_c$
- The carry chain delay through the first Function Block, t_{CARY8}
- the carry lookahead delay for the second Function Block, t_{CARYFB}

Cycle time = $1/f_c + t_{CARY8} + N(t_{CARYFB})$

N is the number of additional Function Blocks.

Using the XC7354-10 values from the data sheet makes the total carry delay 19.5 ns. If the adder is made one to nine bits wider, it crosses one additional Function Block boundary. Consequently the total delay increases by one t_{CARYFB} delay; the 19 to 27-bit adder settles in 21 ns.

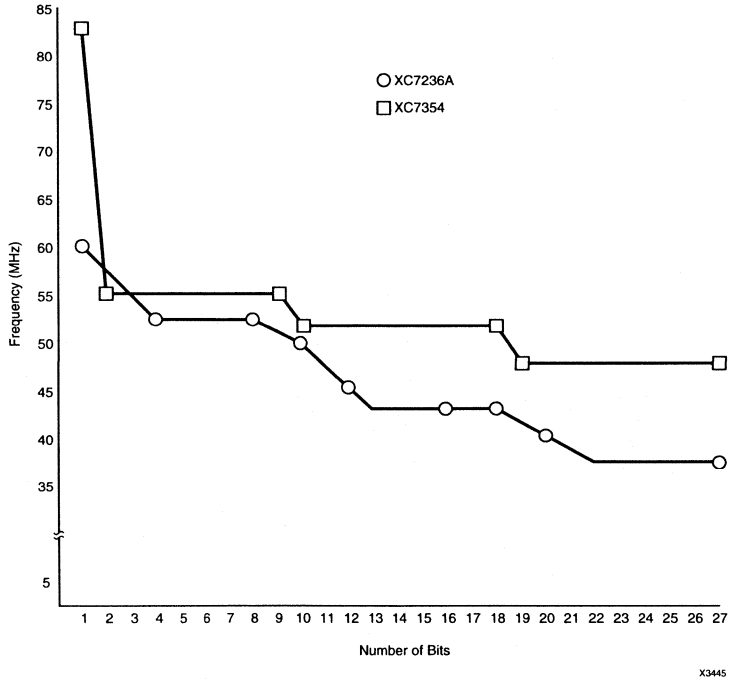


Figure 3. Arithmetic Performance

18-Bit Pipelined Accumulator

XAPP 039.001 Application Note By DAVE GRACE

Summary

This Application Note describes a pipelining technique that significantly improves the throughput of an accumulator.

Xilinx Family

XC7200/XC7300

Demonstrates

High Speed Arithmetic

Introduction

Digital Signal Processing, image processing, and graphics applications require high-performance arithmetic in the data path. The XC7272 can operate as an 18-bit accumulator, running at up to 29 MHz with a pipeline latency of one extra clock, or at 25 MHz without pipeline latency. The pipelined design is described below.

Operation

The incoming 18-bit data word is split into two words of unequal length. The lower ten bits are accumulated immediately, while the higher eight bits are registered in REGH_IN. During the second clock period, the registered carry-out of the lower word and the registered higher 8 bits are accumulated; the output of the lower 10-bit accumulator is pipelined in REGL_OUT.

In many applications, an input register improves system timing. It does, however, introduce an additional pipeline delay.

The design uses 40 of the 72 Macrocells available in the XC7272, and takes advantage of the arithmetic carry and ALU capability in each Macrocell. Input registers can be used for synchronizing the input data. In a conventional EPLD, this design would consume more resources, and would run substantially more slowly.

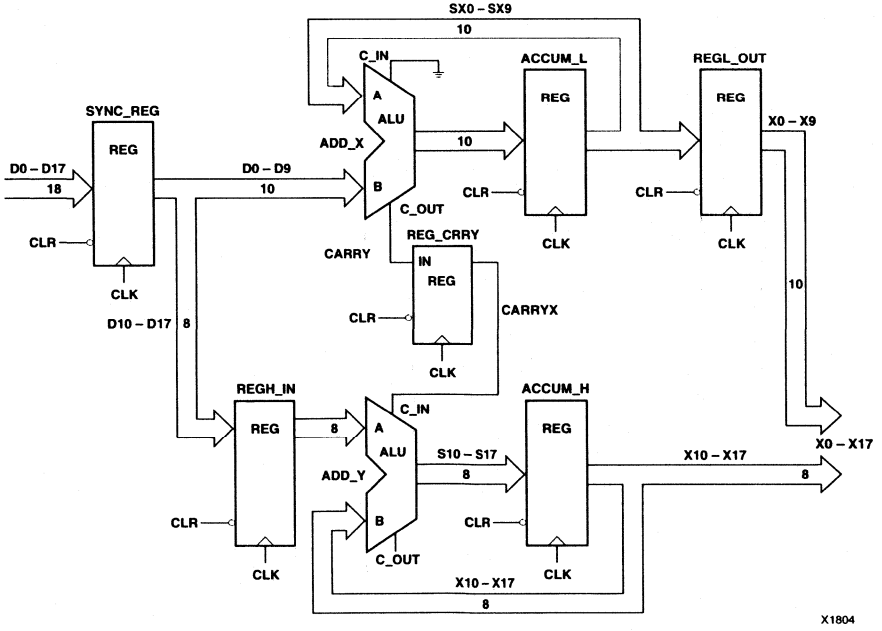


Figure 1. 18-Bit Adder/Accumulator Block Diagram

Register-Based FIFO

XAPP 005.002

Application Note By BERNIE NEW AND WOLFGANG HÖFLICH

Summary

While XC3000-series LCA devices do not provide RAM, it is possible to construct small register-based FIFOs. A basic synchronous FIFO requires one CLB for each two bits of FIFO capacity, plus one CLB for each word in the FIFO. Optional asynchronous input and output circuits are provided. Design files are available for two implementations of this design. The fastest of the two implementations uses a constraints file to achieve better placement.

Specifications		*Xilinx Family*
Size	8 x 8 Bits	XC3000/XC3100
Maximum Clock Frequency XC3100-3	42 MHz	
Number of CLBs	40	

Introduction

In the absence of RAM, XC3000 FIFOs must be constructed with registers. Using both flip-flops, one CLB is required for each two bits of FIFO capacity. For a synchronous FIFO, an additional one CLB per word is required for control. Thus an 8-word by 8-bit FIFO can be implemented in 40 CLBs. Speed is a function of depth, with an 8-word FIFO able to achieve speeds of up to 42 MHz.

Asynchronous inputs and outputs may be added if desired. Each of these adds n/2 CLBs for an n-bit wide FIFO, plus a few additional CLBs for control logic. Typically, asynchronous inputs and outputs operate more slowly because of the handshake required for synchronization. Where burst input or output speed is required for data transfer, the FIFO should be operated in synchronism with the high-speed port.

The basic designs shown use simple flags that permit the input and output of single words. For block transfers, flags could be generated for signaling the availability of a block of data or space for a block of data.

Synchronous FIFOs

The basic FIFO design, shown in Figure 1, comprises a broadside shift register; each word has a separate shift enable. A control flip-flop, associated with each word, contains a valid flag that is shifted with the data. The shift-control logic uses these valid flags to generate shift enables and control the flow of data through the FIFO.

Whenever a register does not contain valid data, shift is enabled for that register, and for all the registers upstream from it. This causes data to continuously shift through the FIFO, with valid words backing-up at the output. They remain there until a POP command enables the shift in all the registers in the FIFO. Invalid data is not retained.

Figure 2 shows the detail of the FIFO. For simplicity, only two data bits are shown (the top two rows of flip-flops); all other data bits are identical. The bottom row of flip-flops contains the valid bits. The shift control logic is the chain of OR gates; a column of flip-flops is enabled if its valid bit, or any valid bit to the right, is not asserted.

The POP command acts like an additional active-Low valid bit, which is to the right of all the columns in the FIFO. When it is High, all the registers shift. If the second to last register contains valid data, this is shifted into the last register, and the VALID flag remains High. Otherwise, invalid data is shifted into the last register, and the VALID flag goes Low. The last register continues shifting until it receives valid data, when the VALID flag goes High.

Data can only be written into the FIFO if the first register contains invalid data or valid data that is about to be shifted out. This condition is signaled by the RDY flag, that is also the shift enable for the first register. Conse-

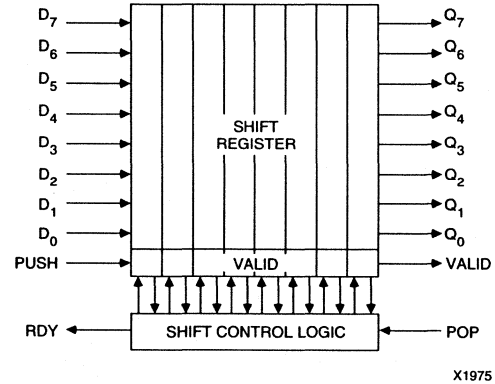


Figure 1. 8-Word x 8-Bit Synchronous FIFO (40 CLBs)

Supporting design files are available on the Xilinx Technical Bulletin Board under the name XAPP005V (Viewlogic)

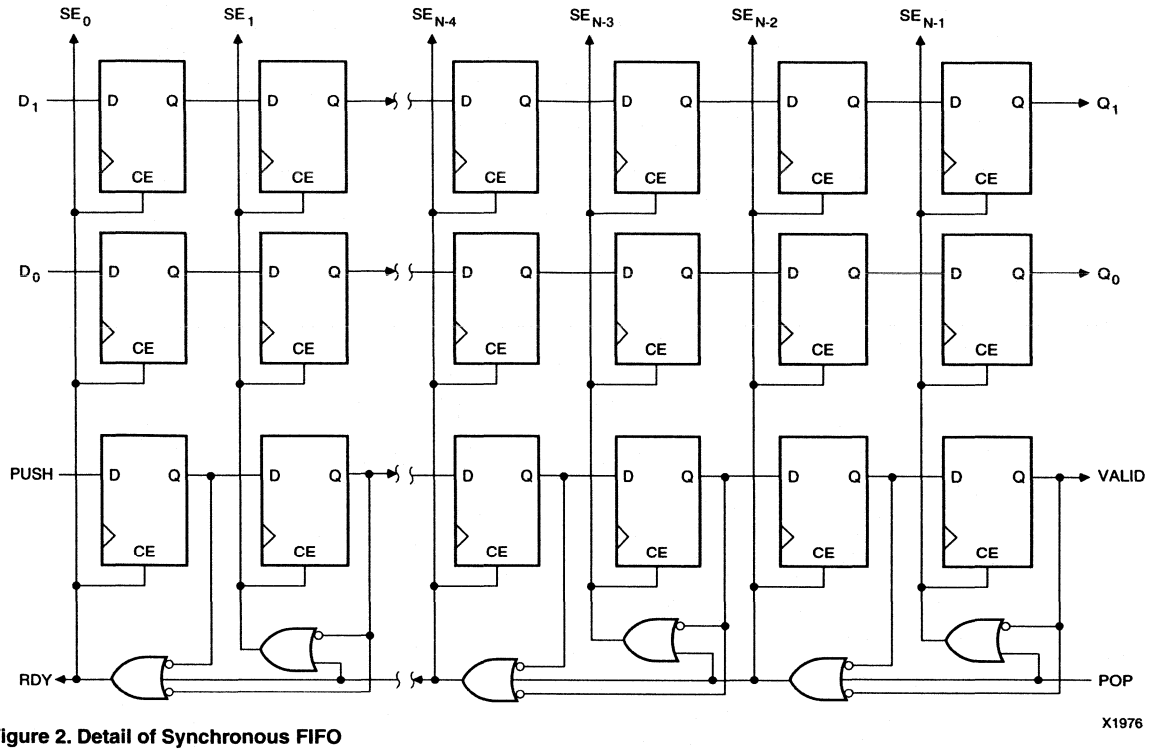


Figure 2. Detail of Synchronous FIFO

quently, data is always being shifted in when the FIFO is ready. The function of PUSH is simply to identify the data being shifted in as valid, so that it is retained in the FIFO.

In the diagram, the CLB clock enable (CE) is used as shift enable. When combining pairs of flip-flops into CLBs, CE can only be used if adjacent bits of the same register are combined. If it is more convenient, bits of equal weight from adjacent registers may be combined. In this case, function generators must be used to implement shift enable. This entails a simple 2-input multiplexer that selects input data when shift is enabled, and selects existing data from the flip-flop when it is not enabled.

The speed of the FIFO is determined by the ripple-OR time of the shift-control logic, and the distribution and set-up times of the shift-enable signals. This defines the set-up time for the POP command. The settling time for the shift-control logic is one CLB delay per two words of FIFO depth. Longlines should be used to distribute the shift-enable signals.

Asynchronous Input Stage

Asynchronous data may be entered into the FIFO using the circuit shown in Figure 3. An additional input holding register is provided to facilitate edge-triggered input. If appropriate, this can be implemented in IOB registers.

Data may only be entered when the RDY flag signals that the input register is available to accept it. The input clock (PUSH) also asserts the PUSH INP signal which removes the RDY flag. On the next internal clock, PUSH INT is asserted and PUSH INP cleared. When shift is enabled into the first register of the FIFO, data is transferred out of the holding register, PUSH INT is cleared and RDY is re-asserted.

If data is being input from a synchronous system that is not synchronized to the FIFO internal clock, the circuit shown in Figure 4 should be used. Again, an input holding register is provided. However, it is enabled by PUSH, instead of being clocked by it (an IOB register cannot be used). As before, PUSH causes PUSH INP to be asserted. Feedback around the flip-flop sustains PUSH INP until it is recognized by the internal clock, permitting the PUSH command to be removed after the one input clock.

The entry of data into the FIFO proceeds as in the previous scheme. RDY is registered to synchronize it to the input clock. The negative clock edge is used for this, so

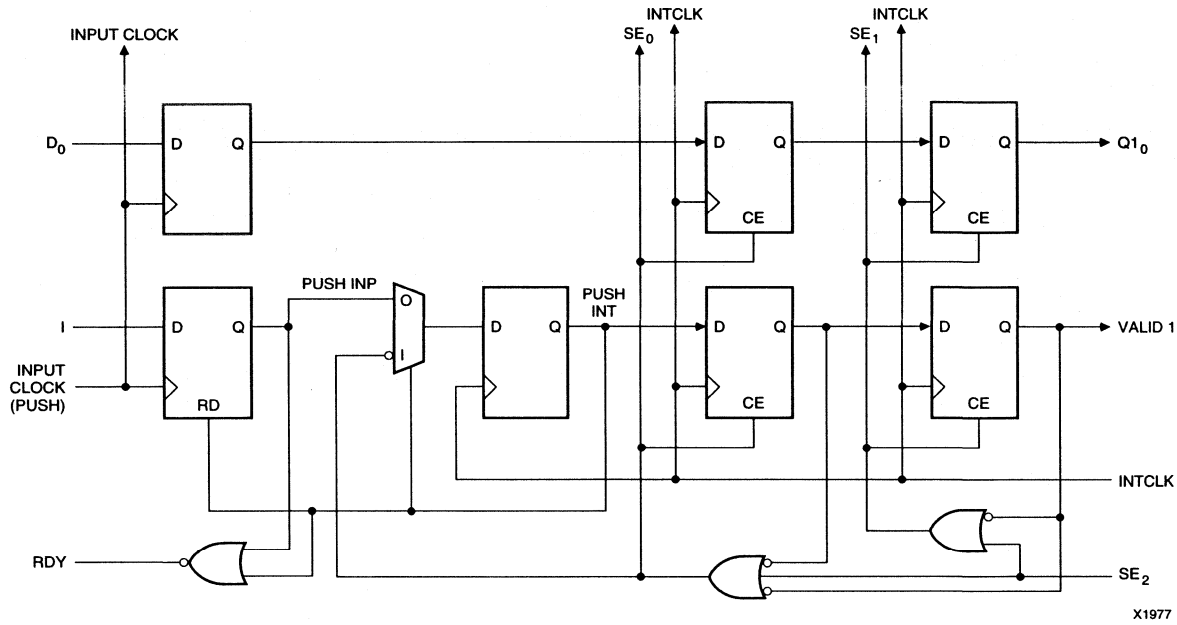


Figure 3. Asynchronous Input Stage

that, if the FIFO is sufficiently fast and is not full, the RDY flag will remain set, and data can be entered on successive input clocks. If the positive clock edge had been used, RDY would always be Low for at least one clock. At best, this would only permit data to be entered on alternate input clocks, no matter how slow.

Asynchronous Output Stage

The circuit shown in Figure 5 should be used, if an asynchronous output is required. For an immediate, edge-triggered output, a holding register is provided, which is clocked by the output clock (POP). IOB flip-flops may be used for this register.

The output register may only be clocked when the RDY flag signals that data is available in the last register of the FIFO. The output clock causes data to be transferred out of the FIFO, and asserts POP OUT. This removes the RDY flag. On the next internal clock, POP INT is asserted and POP OUT is cleared. POP INT is held, and the FIFO shifts, until the last register again contains valid data. It is then cleared, and the RDY flag is re-asserted.

If data is being output to a synchronous system that is not synchronized to the FIFO internal clock, the circuit shown in Figure 6 should be used. The output register, which cannot be implemented in IOBs, is enabled by POP. POP also causes POP OUT to be asserted. Feedback around the register sustains POP OUT until it is recognized by the internal clock, even if POP is removed and another output clock occurs.

The transfer of data out of the FIFO proceeds as in the previous scheme. RDY is synchronized with the negative edge of the output clock. As a result, data can be output on successive clocks if the FIFO is fast enough and data is available.

Implementation Notes

The obvious organization for the FIFO is as a rectangular array of CLBs, with the control logic in the bottom row. The flip-flops may be partitioned into CLBs in two ways. If adjacent bits of the same word are combined, the result is a FIFO that is twice as wide as it is tall (assuming equal numbers of bits and words).

Alternatively, two bits of equal rank from adjacent words may be combined. This gives a FIFO that is twice as tall as it is wide and is potentially faster. The critical path through the control logic passes through a chain of half as many gates as there are words. The tall, narrow organization allows these gates to be implemented in adjacent CLBs with zero-delay direct interconnects.

Both forms of the FIFO are available as macros, using CLBMAPs.

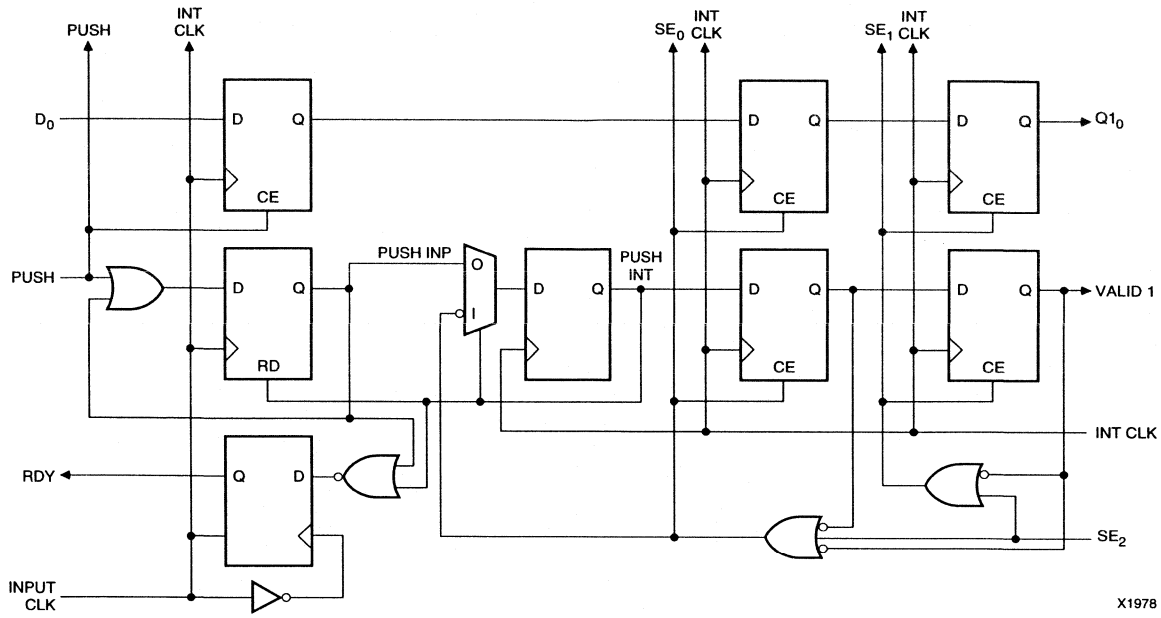


Figure 4. Asynchronous Input Stage (From Synchronous System)

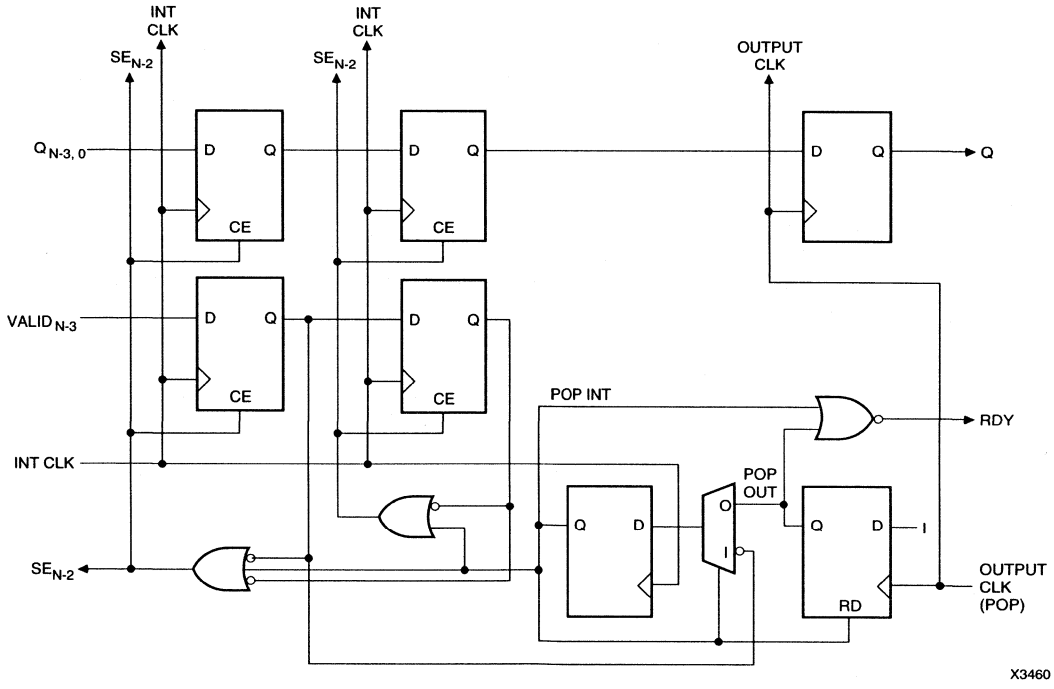


Figure 5. Asynchronous Output Stage

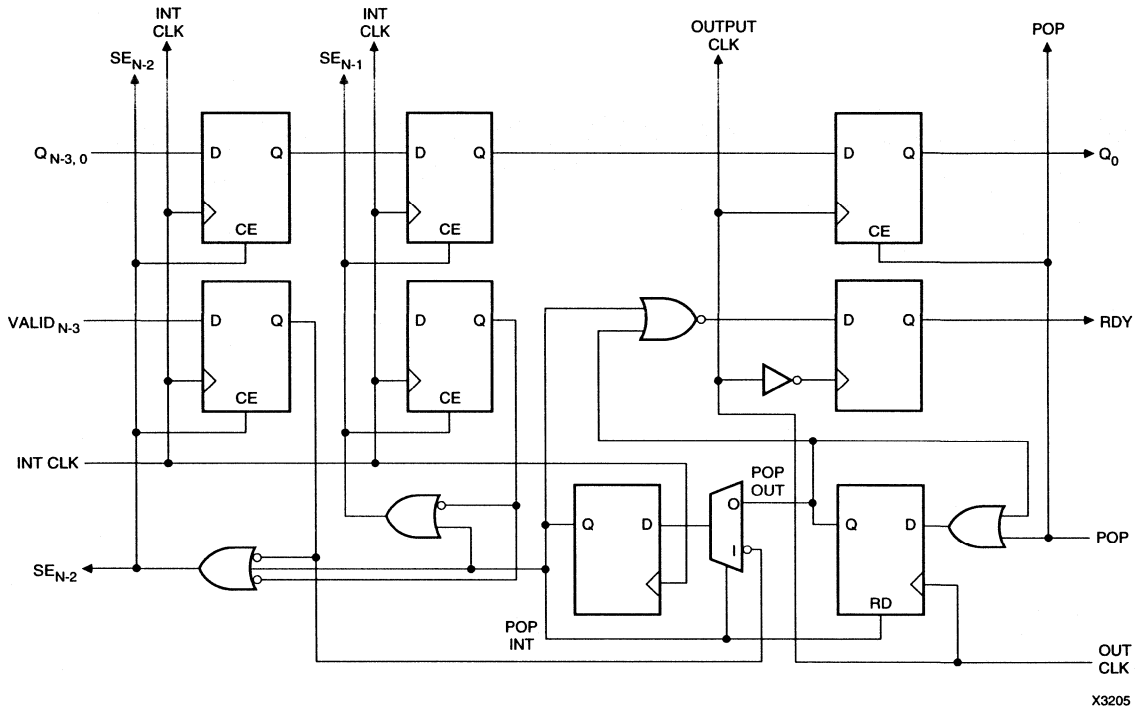


Figure 6. Asynchronous Output Stage (To Synchronous System)

Using the XC4000 RAM Capability

XAPP 031.000

Application Note By ROMAN IWANCZUK

Summary

The XC4000 family of LCA devices permits CLB look-up tables to be configured as user RAM. This Application Note provides background information for users of the feature, and points out the need for carefully designed control logic. The Application Note, *High-Speed RAM Design in XC4000* (XAPP 042, page 8-139), shows a rugged, simple and elegant solution to this problem.

Xilinx Family

XC4000

Demonstrates

XC4000 RAM Capability

Introduction

LCA devices emulate logic using a look-up-table-based architecture. The look-up tables are implemented in static RAM that is written during configuration and read during operation. Unlike previous LCA families, the XC4000 family also permits the RAM to be written during operation. Using this feature, internal RAM can be included in user designs.

The RAM function in the XC4000 permits a significant increase in the system functionality that can be implemented in an FPGA. This includes traditional RAM-based logic such as FIFOs, LIFOs, register files, as well as novel applications such as the RAM-based shift register described in this application note. Interfacing to the RAM is not particularly difficult, but it requires an understanding of the issues involved.

With ~10 ns cycle time, the XC4000 RAM is much faster than the memories with which most designers are familiar; most discrete SRAMs have cycle times of 55 ns or longer. Consequently, the design of XC4000 control circuitry is more critical. Many factors, such as interconnect delays, that can usually be ignored in discrete RAM designs cannot be ignored in an LCA RAM design. Using the XC4000 RAM is like using very fast discrete SRAMs (<25 ns cycle time), where similar factors must be considered.

Figure 1 shows the address, data and control signals available on the XC4000 RAM compared to those of a discrete SRAM. Notice that the output of XC4000 RAM is permanently active, since it does not have a Chip Enable control. If a 3-state output is required the Data-Out signal can be connected to a TBUF input, as described later. Another difference is that the Write Enable on the XC4000 RAM is active High, while it is typically active-Low on discrete SRAMs. Some functional differences also exist, and are described later in this section.

A further point to note is the granularity of the XC4000 RAM. The example shown is a 16 x 1 memory, the smallest XC4000 RAM primitive. A similar 32 x 1 primitive exists; both these primitives can be combined to provide larger memories. In contrast, the smallest monolithic SRAM used in today's designs is generally 4K bits.

XC4000 RAMs consume CLB resources that could otherwise be used to implement logic, and large RAMs may restrict the amount of logic that can be included. Table 1 shows the resources used by each of the RAM primitives, and how many of each could implemented if various members of the XC4000 family were entirely devoted to RAM. As may be seen, the total amount of memory that can be implemented in an entire XC4010 is only 12,800 bits, making it a very inefficient replacement for large SRAMs

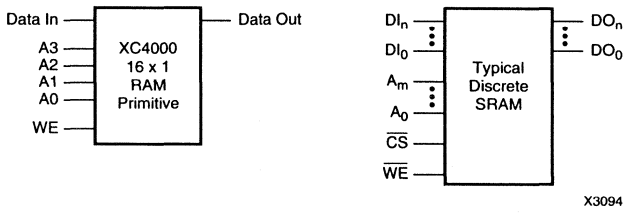


Figure 1. XC4000 RAM and Discrete SRAM Connections

Table 1. Trade-off Between RAM Primitives and Logic.
Note: If all CLB's are used as RAM there are none available for logic implementation

		Maximum Number of RAM Modules		
RAM Module	**Equivalent Logic**	**XC4003**	**XC4005**	**XC4010**
16 x 1	4-input Function Generator (F or G)	200	392	800
32 x 1	Two 4-input Function Generators and One 3-input Function Generator (F+G+H)	100	196	400

The XC4000 RAM is intended for use in small, fast RAMs in applications like FIFO buffers, scratch-pad memories and register files. For applications that require larger RAMs, it is generally more cost effective to use an external monolithic SRAM connected to the XC4000. This would, however, increase the number of I/O pins needed on the FPGA, and potentially decrease the speed of the design due to the off-chip memory accesses.

Figure 2 shows the read-cycle timing of the XC4000 RAM compared to that of a discrete SRAM. For the comparison, the SRAM is executing an address-controlled read cycle, where the Chip Select signal is permanently asserted, since the XC4000 RAM primitives do not have Chip Select control. The Write Enable signal is not shown in these diagrams, and must be remain inactive during a read cycle in both cases. The diagrams are not drawn to scale to permit the relative shapes of the waveforms to be compared more easily.

The diagrams are very similar. The only difference is that on the discrete SRAM, the output data cannot change for a period, t_{OH}, after an address change, while it can change immediately in XC4000 RAM. This parameter is not specified explicitly, but the output of any CLB must be considered invalid immediately following an input change. This is not a problem in most designs.

The corresponding write-cycle comparison is shown in Figure 3. To match the XC4000 RAM, the SRAM timing is for a Write-Enable-controlled write cycle, where the Chip Select signal is permanently asserted. Again, the diagrams are not drawn to scale so that the relative shapes of the waveforms can be compared easily.

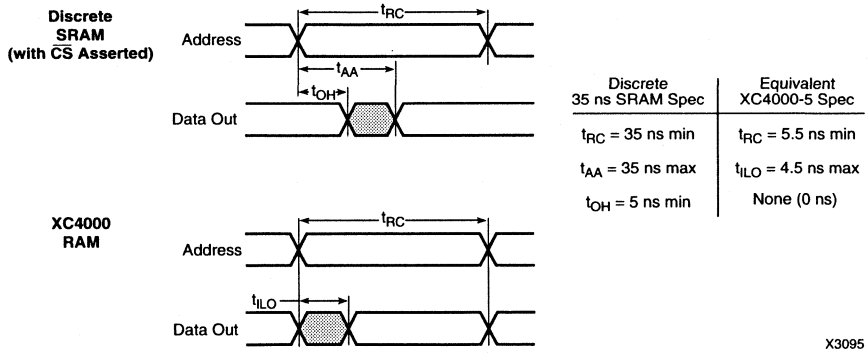
The primary difference between the discrete SRAM timing and the XC4000 RAM timing is the address hold parameter (t_{WR} on the SRAM, t_{DH} on the XC4000). In the XC4000, the address **must** remain stable for 2 ns after the Write Enable signal has been removed. This difference significantly impacts the design of the control logic, as will be discussed later.

While the Data Out signals are not shown in these diagrams, these, too, are different. In most discrete SRAMs, the Data Out signal is high impedance during the Write-Enable-controlled write cycle. In the XC4000 RAM, however, the data output has no high-impedance state and, therefore, remains active.

The write cycle starts by reading the existing data in the location addressed, and then, after WE is asserted, changes to reflect the new data. For the exact timing data output signal, please refer to the timing diagram "Read during Write" in the XC4000 data sheet.

Potential Control Logic Problems

As in any XC4000 design, the primary concern in a RAM design must be to meet the worst-case timing requirements described in the data sheet. Failure to do so can result in a design that appears to work perfectly correctly



Discrete 35 ns SRAM Spec	Equivalent XC4000-5 Spec
t_{RC} = 35 ns min	t_{RC} = 5.5 ns min
t_{AA} = 35 ns max	t_{ILO} = 4.5 ns max
t_{OH} = 5 ns min	None (0 ns)

Figure 2. XC4000 RAM and Discrete SRAM Read Cycles

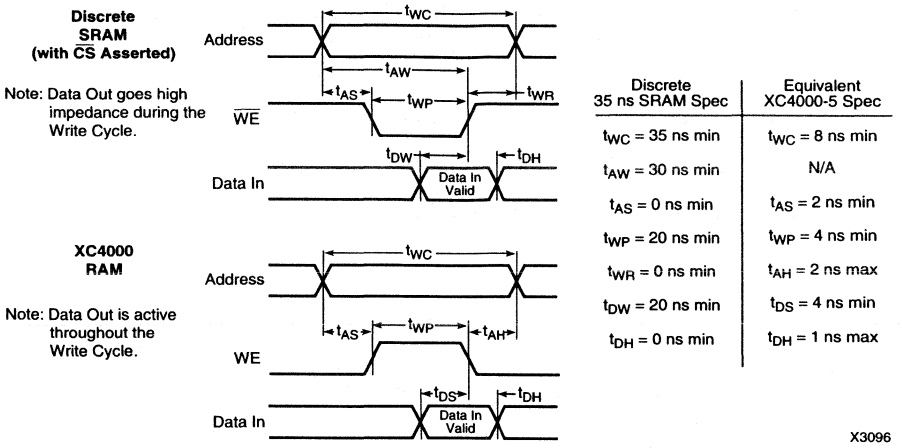


Figure 3. XC4000 RAM and Discrete SRAM Write Cycles

on the bench, but fails at a temperature extreme, at a V_{CC} extreme, or with a device from a different production lot.

A second area of concern is signal glitches, which must be avoided at all costs. Two types of glitches can cause problems in any SRAM-based design: glitches on the WE line and glitches on the address lines while WE is asserted. As has been stated earlier, the XC4000 RAM is extremely fast, and even glitches that do not meet the minimum specification for guaranteed operation can disrupt the contents of the RAM. The areas of primary concern are shown in Figure 4.

Figure 5 shows an example of a glitch-generating control circuit that might be used to generate the WE pulse in a FIFO. Notice in the timing diagram that the WE pulse is generated perfectly when Q0 and Q1 are both High. The glitch can occur as Q0 changes from 1 to 0 and Q1 "simultaneously" changes from 0 to 1; if Q1 changes before Q0, there is a momentary state that meets the requirements of the AND gate to generate WE.

This circuit might be adequate in a discrete RAM design. By judicious choice of components, the minimum timing specifications of the counter and the AND gate could be matched to ensure that glitches do not occur. Such

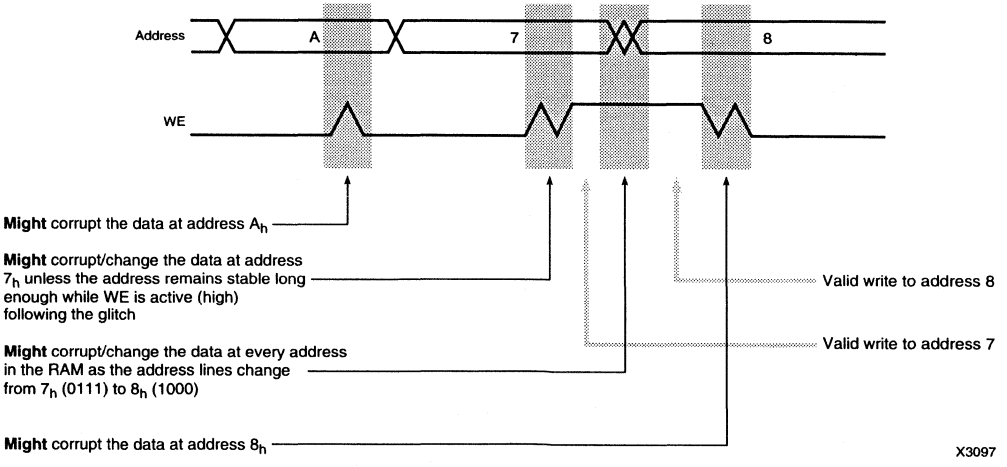


Figure 4. Typical Glitches That May Cause an XC4000 RAM Design to Fail

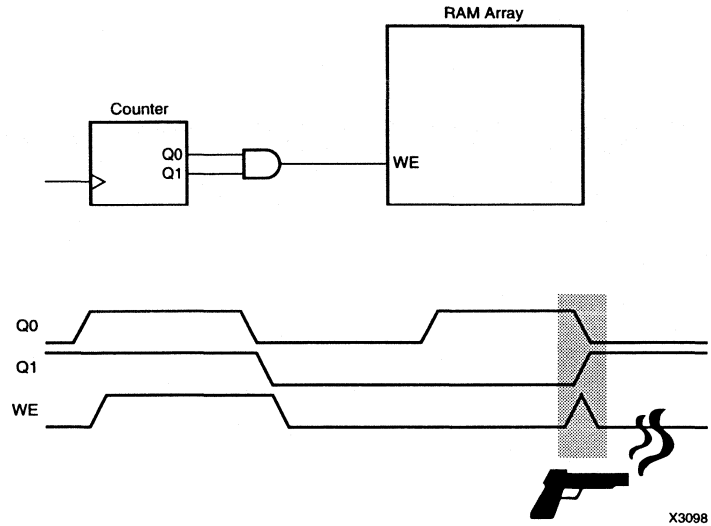


Figure 5. Example of a Marginal WE Generation Circuit

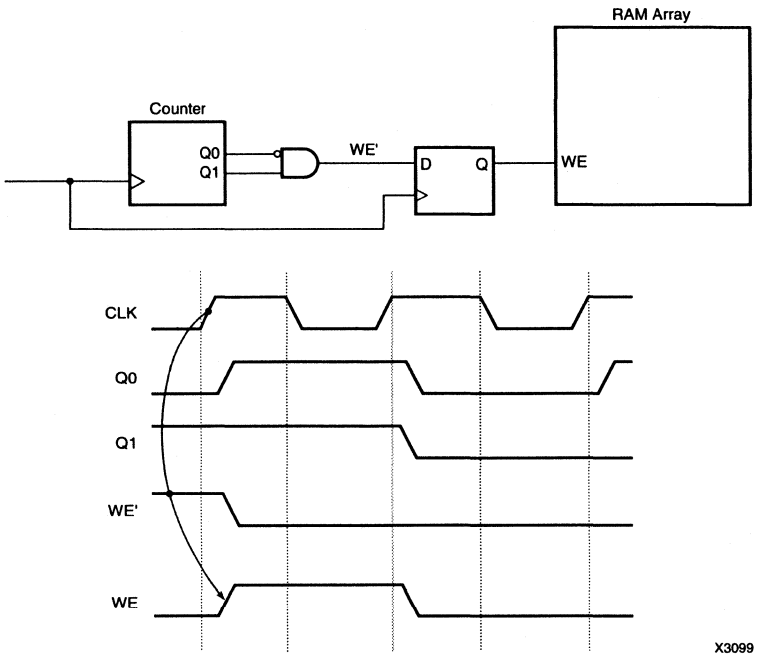


Figure 6. Example of a Glitch-Free WE Generation Circuit

matching is not possible in an FPGA environment, where the possibility of glitches is increased by the high speed of the logic functions and the relatively long routing delays. Figure 6 shows a better, glitch-free circuit.

In general, some valid techniques used in a discrete design can create marginal designs in the high-speed LCA environment. Avoid asynchronous circuits like the plague. With a little thought, most things that are be done asynchronously can be better done synchronously. If necessary, use small Gray-code or Johnson counters that can be decoded in a hazard-free manner. In a Xilinx FPGA, such counters are as easy to implement as binary counters.

Routing Delay Issues

FPGA routing delays can cause a circuit that works at speed on paper not to operate under worst-case conditions. In this situation, worst-case conditions must be interpreted as slow operation, fast operation, or any combination of these that causes a malfunction. The following issues should be considered.

- The WE signal is skewed in time by the routing delay introduced by its net. Make sure that the circuitry used to control the address and data signals takes this into account. The t_{AH} and t_{DH} requirements must not be violated.
- Compared to small RAM arrays, large RAM arrays have higher fan-out address lines with longer routing delays. Consequently, for a given speed, the address-generation circuits have less time in which to operate. Generally, large, fast RAM arrays require more ingenious control circuitry, and may necessitate partial duplication of the address circuitry to drive separate segments of the RAM array.
- RAM modules which need to run at speed benefit greatly from manual placement. It pays to create a trial design that only implements the RAM and its control logic. This small design can be quickly placed and routed, and then optimized in the XACT Design Editor (XDE). The optimized placement can the be incorporated into the main design using location constraints. Alternatively, the RAM portion can be converted into a hard macro, thus preserving its relative placement.

Creating a RAM Array

The XC4000 RAM is accessed as 16 x 1 and 32 x 1 primitives. In RAM applications requiring less than 16 words, 16 x 1 modules must be used with any unused addresses tied to ground or V_{CC}. 16 x n and 32 x n arrays can easily be created by connecting several of these primitives in parallel with common address signals.

For depths greater than 32 words, a RAM array must be constructed as shown in Figure 7. In this example, two 32 x 1 primitives are combined to implement a 64 x 1 RAM. The most significant address bit is used to select between the primitives, while the remaining address bits are common to both. During a read cycle, selection between the primitives involves multiplexing the output data. For a write cycle, the data is common to both primitives, and the WE pulse is gated to enter the data into only one.

TBUFs could be used to create the output multiplexer. However, at least half of a horizontal Longline would be

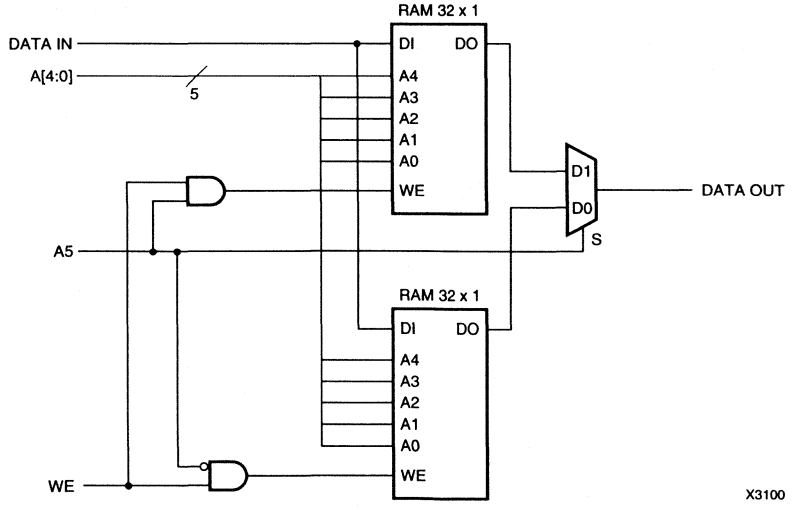


Figure 7. 64 x 1 RAM Array

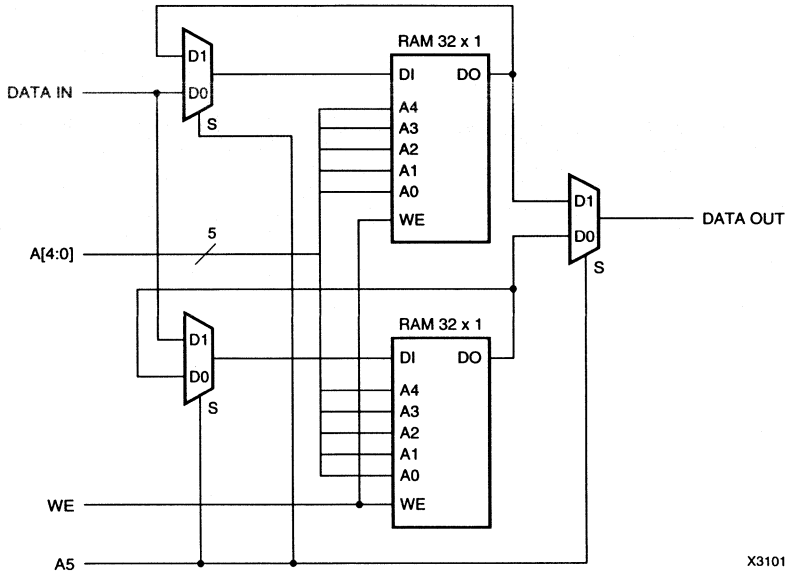


Figure 8. Alternative 64 x 1 RAM Array

consumed for each bit of RAM width, and TBUFs are slower than small logic-based multiplexers. Consequently, the use of TBUFs is only recommended for very deep RAM arrays.

Gating the WE pulse increases the delay in the WE path. This delay is not usually a problem in slower RAM applications; but, as the write-cycle time decreases, the additional delay can become unacceptable.

Figure 8 shows an alternative technique. While new data is being written into the selected primitive, the existing data is re-written into the non-selected RAM primitive. This technique introduces additional delay into the data input path, but maintains the minimum delay in the WE path, which is often the critical path. The circuit choice depends on the timing requirements of the specific system.

These expansion techniques are directly analogous to depth expansion in discrete RAMs. The only differences are the explicit output multiplexer, which would be implemented using 3-state busses in the discrete case, and the Write Enable gating, which is integrated into discrete RAMs.

Emulating SRAM with Bidirectional Data Pins

Some commercially available discrete SRAMs have a single Data Input/Output pin. This type of SRAM can be emulated in the XC4000 using the circuits shown in Figure 9. In Figure 9a, the multiplexing is performed using IOB elements; the signals inside the FPGA are unidirectional.

In Figure 9b, the bidirectional data line is extended into the FPGA and the RAM uses TBUFs to drive the data line. This circuit is appropriate where multiple data sources are required to read/drive the data line at different times.

Note that in Xilinx FPGAs, the 3-state buffers (TBUF, OBUFT) have enable signals that are active-High 3-state controls, i.e., when a logic 1 is applied, the output of the buffer is high impedance, and when a logic 0 is applied, the output is active. The T pins can be viewed as active-Low Output Enables.

Recommended Control Logic Schemes

There are many ways to generate the WE signal for the XC4000 RAM. The choice is design dependent, and a major factor is whether the design is synchronous or asynchronous. In an asynchronous design, the WE pulse is generated from a signal originating outside the FPGA that may be gated with internally generated signals. In a synchronous design, the WE pulse is generated by logic that is completely within the FPGA.

Asynchronous Control Logic

In the asynchronous case, each design will be different, and depend on the external signals that are available. Consequently, it is impossible to make firm recommendations. However, the following discussion should illustrate some basic techniques.

Asynchronous designs generally take the form shown in Figure 10. External signals from an interface, usually a

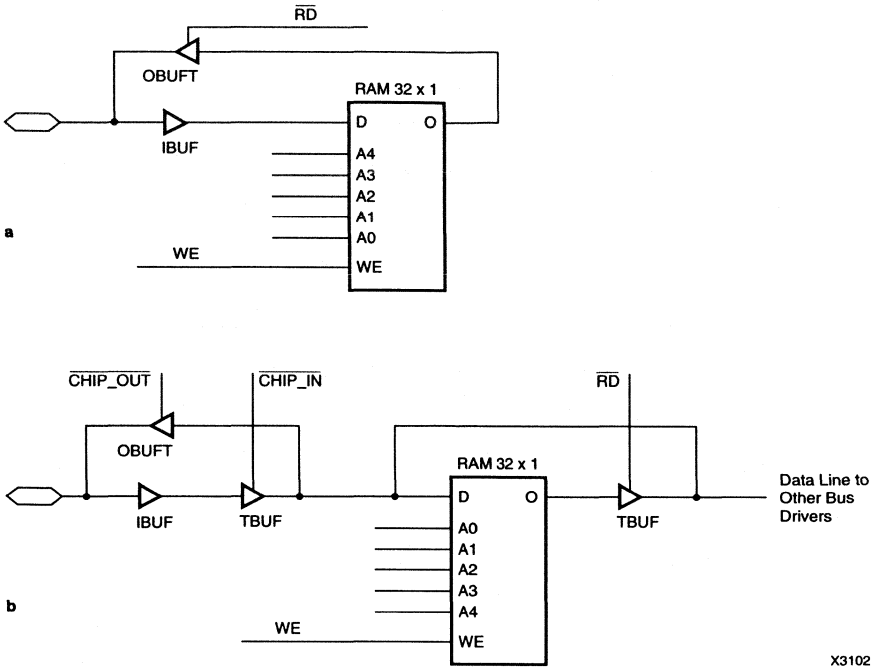


Figure 9. Methods of Emulating a RAM that has Bidirectional Data Pins

microprocessor bus or a system backplane, are used to generate the address, control and data to the RAM. Typically, the designer is required to combine input signals to control the RAM. While bus transfers are often fast, the read cycle is usually not a problem; it is the write cycle that is difficult.

The biggest problems facing the designer are the following.

- How to create a WE signal that, at the same time, is compatible with the data and address timing of the system bus and meets the set-up and hold-time requirements of the RAM.
- How to create such a WE signal with no glitches.

Solving these problems requires creativity. The following example describes how to solve a typical problem.

Figure 11 shows the system timing for the read and write cycles of a typical microprocessor. As mentioned previously, the design of the read-cycle control logic is rarely a problem, since the necessary interface signals are usually present early in the cycle. All that needs to be generated is a subset of the address for the RAM, and an enable signal to the output drivers. This can be done using the circuit shown in Figure 12.

The XC4000 wide decoders can be used to generate an address valid signal that can be gated with other interface control signals. The resulting signal indicates whether the RAM is being addressed during the current cycle. If the current bus cycle is a read, this signal should be registered by a flip-flop on the rising edge of T2. The resulting Qualified Read signal is used to enable the output buffers. The portion of the microprocessor address routed to the RAM depends on the size of the RAM.

The write-cycle timing can be generated similarly to the read-cycle timing, except that the flip-flop generating the Qualified Write signal would have its CE pin connected directly to the W/R# signal. This is shown in Figure 13, which also shows the timing of the Qualified Write signal.

The write-cycle timing is more difficult that the read-cycle timing, because both the address and data hold times must be met, even with worst-case timing. It may be necessary to register the address or data to extend the time during which they are stable, Figure 14. The falling edge of the ADS# signal is used to register the address lines driving the RAM. Note that the address lines used in the control logic gating should not be registered. This would make it difficult to meet the set-up times of the flip-flops that generate the Qualified Read and Qualified Write signals.

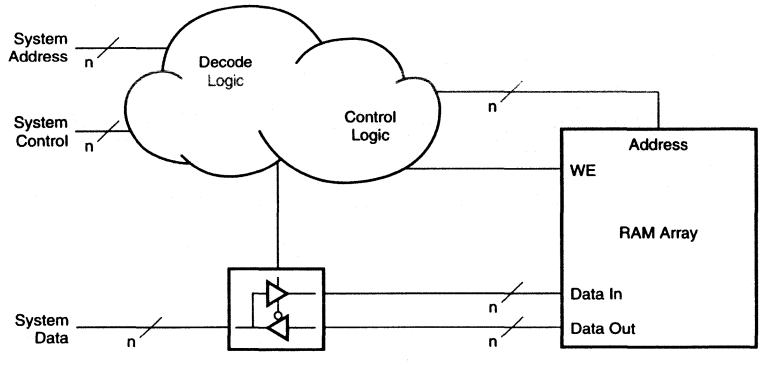


Figure 10. General Form of an Asynchronous RAM Interface

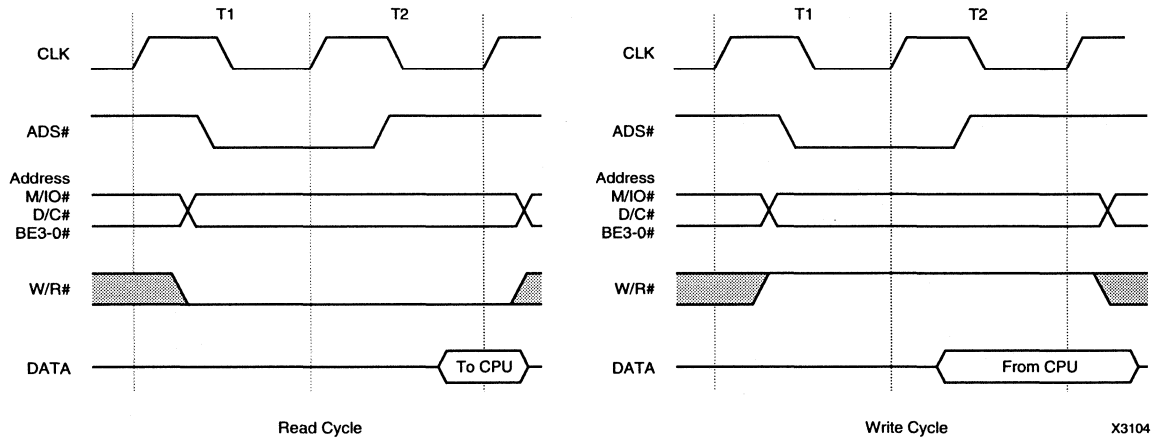


Figure 11. Typical Microprocessor Read and Write Cycles

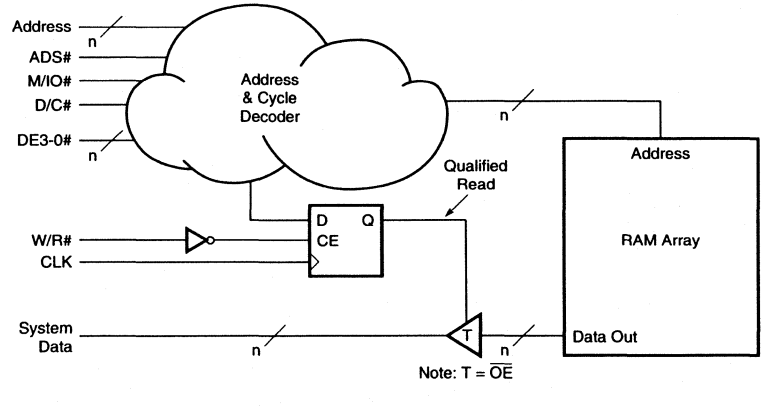


Figure 12. Implementation of the Read Cycle Logic for an Asynchronous Interface

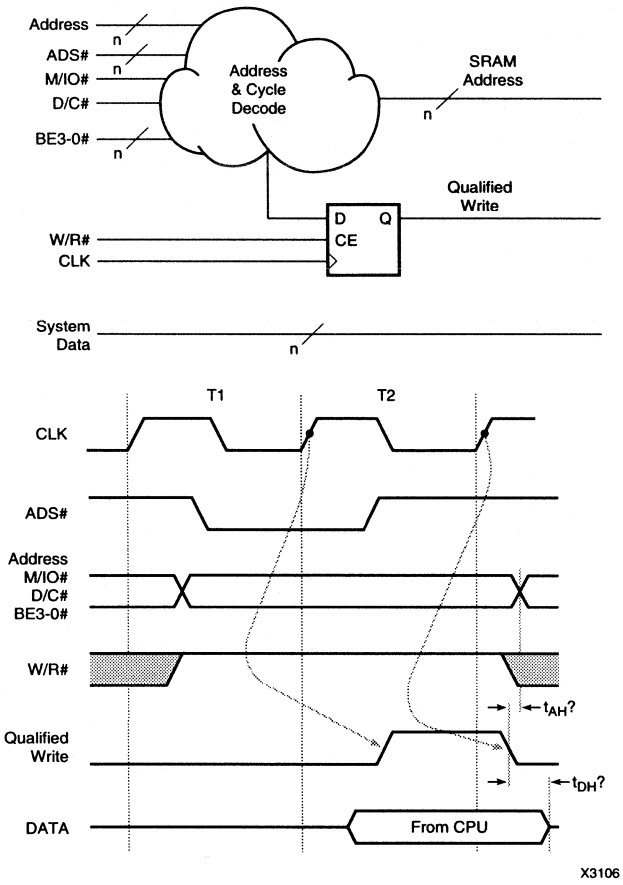


Figure 13. Write Cycle Control Logic and Timing

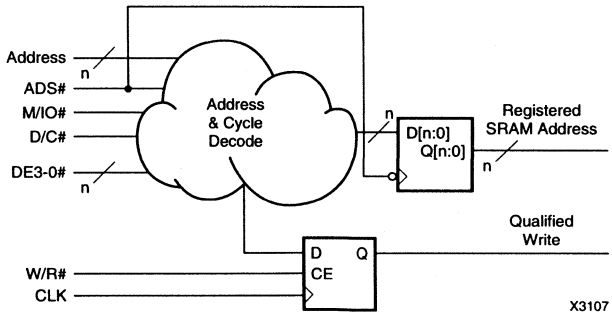


Figure 14. Modified Write Cycle Logic

Synchronous Control Logic

In a typical synchronous RAM application, there is some external stimulus which triggers a read or a write cycle. In response to this stimulus, logic inside the LCA device generates the control signals for the RAM cycle. Potentially, none of these control signals may be derived from external signals; all of them must be generated internally.

Figure 15 shows a 64-bit shift register implemented using RAM. A flip-flop-based 64-bit shift register would use all the flip-flops in 32 CLBs; the RAM-based version can be implemented in only 9 CLBs, a considerable saving of resources. Essentially, the shift register is implemented as a simple circular FIFO that is 1-bit wide and 64-bits deep. To implement a shift cycle, the address pointer is incremented to point to the oldest data in the RAM. Data is read out, and new data is written into the same location. This new data will be read when the address pointer returns to the same location 64 shift cycles later.

The core of this design is a small sequencer that includes the circuit shown in Figure 16. This circuit, when triggered, generates a sequence of four glitch-free pulses corresponding to four successive half periods of the clock, Figure 17. These pulses are used to control the sequence of events required for a shift cycle. The complete waveform diagram is shown in Figure 18.

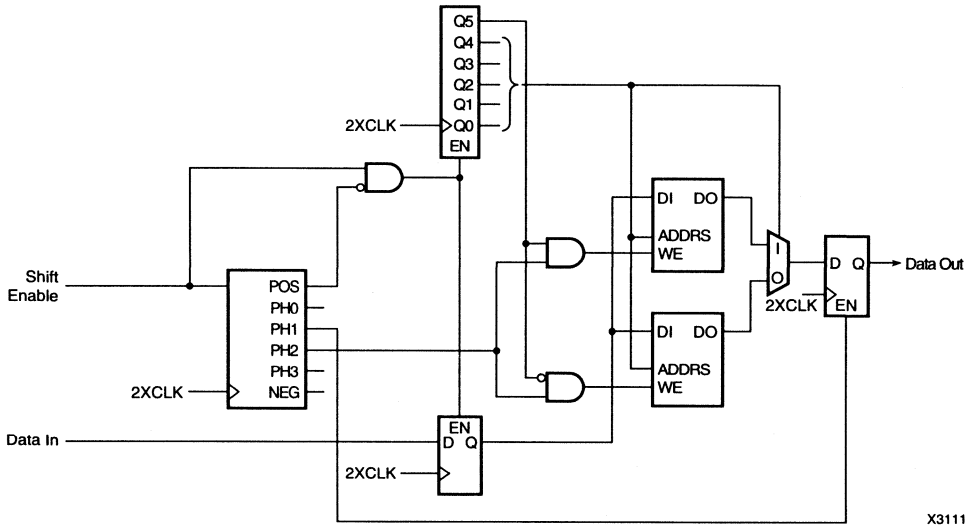


Figure 15. 64-Bit RAM-Based Shift Register

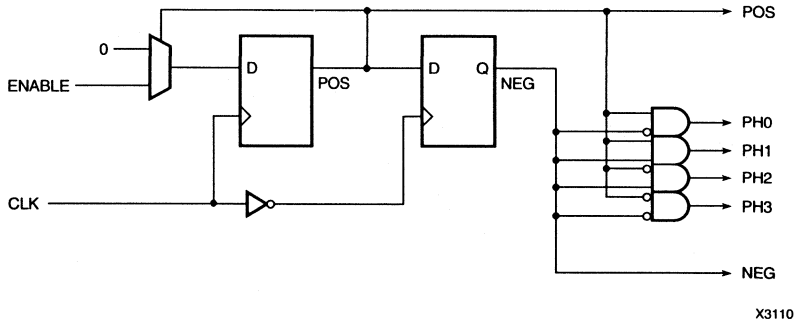


Figure 16. Glitch-Free Sequencer

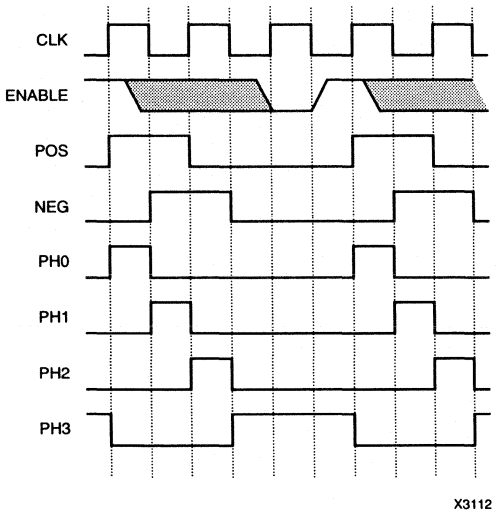


Figure 17. Waveforms for the Glitch-Free Sequencer

The important events are as follows.

1. A shift cycle is initiated on the rising edge of the 2 X CLK by asserting Enable High. At this time, the pulse sequencer is triggered, the input data is captured into a register and the address counter is incremented. This action may occur on any rising clock edge, but is ignored on the rising edge immediately following a trigger.
2. The data written to the RAM 64 clocks previously is read, and is captured into an output register on the rising clock edge that initiates PH2. Both data and address have had a full 2XCLK period to set up. The 0 ns hold time requirement of the CLB is guaranteed, since the data is stable until the WE pulse.
3. New data is entered into the RAM by the WE pulse, which is PH2 delayed by logic and routing.
4. Address and data cannot change until the end of PH3. At least half a period of the 2XCLK is available for to remove of WE and satisfy the address and data hold-time requirements.

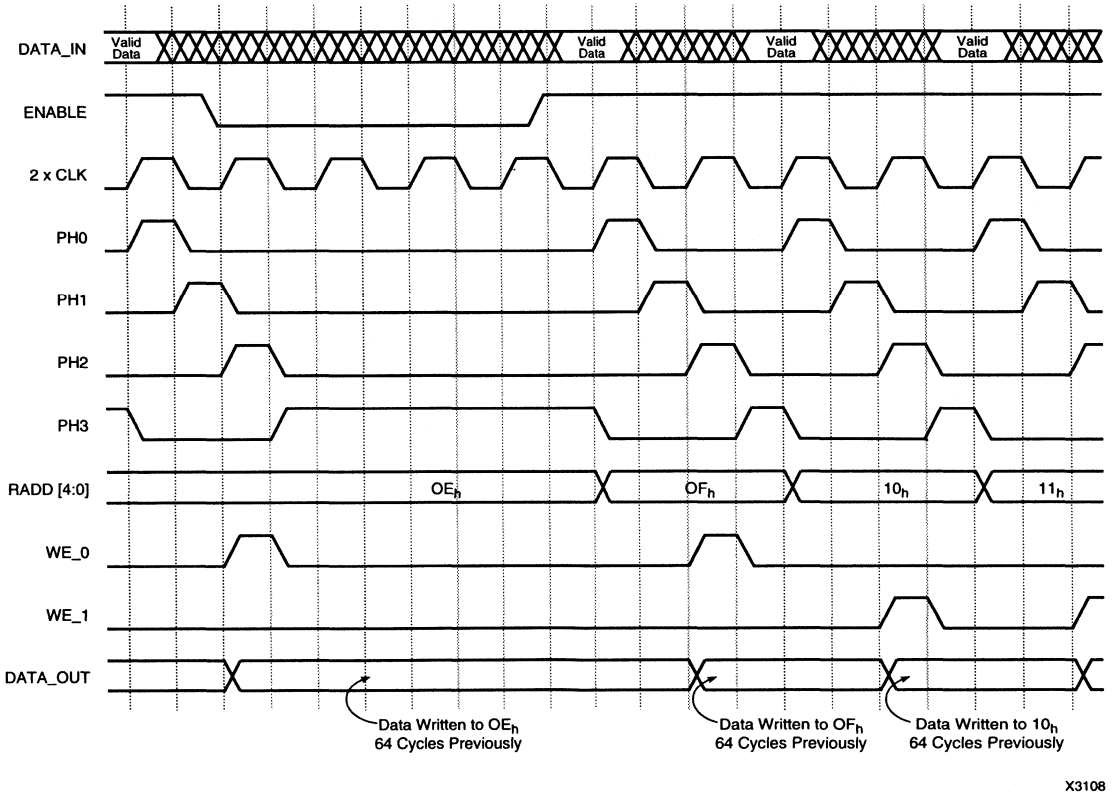


Figure 18. Waveform for Several Shift Cycles of the 64-Bit Shift Register

As can be seen, the pulse circuit allows the orderly sequencing of the write cycle spacing out the events so that timing requirements can be satisfied. This type of sequencing is the preferred technique in synchronous RAM applications. Its advantage is that it is bulletproof; its disadvantage is that it requires a clock that is twice as fast as the cycle time.

The clock does not necessarily need a 50% duty cycle. In the shift-register example, the only duty-cycle restrictions are that the clock High time must generate an adequate WE pulse, and the clock Low time must allow the WE pulse to be removed with sufficient margin to meet the necessary hold times. Within these restrictions, an asymmetrical clock might even be beneficial, providing faster operation.

The Last Resort.

This last solution to the problem is not a nice one, but it works – most of the time. While its operation is not guaranteed by device characterization, the solution almost invariably works at room temperature, with nominal power supplies on typical parts. However, the probability of failure increases as the restrictions are relaxed.

The use of this method in a production design is particularly risky. While it will probably work reliably, occasional failures must be expected due to parts that are close to their specification limit. Additionally, to avoid field failures, every unit should be tested over the full range of temperature and voltage that it is expected to encounter.

Contrary to the advice given earlier, this solution uses an asynchronous circuit to generate a WE pulse, Figure 19. In previous sections, this circuit would have been referred to as a glitch generator, but here it is a pulse generator; that is why it is the last resort!

Using this circuit, the only signal that is needed to perform a write to the RAM is a 1x clock at the RAM cycle rate. The leading edge of this clock sets the data and address, while the trailing edge triggers the WE pulse. The restrictions on the clock are that the address and data must set up during the first half of the clock. The second half of the clock must guarantee the WE pulse time to complete, *at the RAM*, with adequate margin to meet the address and data hold-time requirements.

The pulse-generater circuit is a self-resetting flip-flop. The worst-case loop time is >17 ns on an XC4000-5 device ($2 \times t_{ILO}$) + t_{RIO} + Routing). On the same device, the WE pulse requirement of the RAM is 4 ns minimum. Within a single FPGA, the speed of different logic resources tracks reasonably well (to within 70%). Consequently, the worst-case scenario is the WE pulse width decreasing to 12 ns, while the RAM continues to require a 4 ns pulse. In a faster device, with higher V_{CC} or at a lower temperature, the width of the WE pulse will decrease; but so will the WE requirement of the RAM. As a result, the pulse width should never fail to satisfy the WE requirement.

For more reliable timing, this circuit could be converted to a hard macro in a single CLB. It could then be instantiated in the design as required.

Please see the Application Note, *High-Speed RAM Design in XC4000* (XAPP 042, page 8-139), for a rugged, simple and elegant high-speed RAM design.

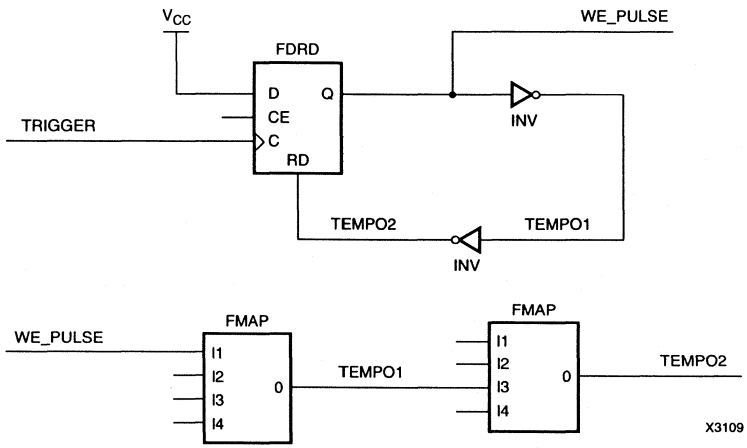


Figure 19. Pulse Generator

High-Speed RAM Design in XC4000

XAPP 042.000

Application Note By BRUCE NEWGARD and BERNIE NEW

Summary

A read-modify-write technique permits the RAM facility in XC4000 LCA devices to operate faster than with conventional read/write operation. In addition, safe operation is guaranteed using a clock at the RAM-cycle rate. As a design example, the implementation of a shift register is described.

Specifications

Minimum Cycle Times (estimated for XC4000-5)

16 x 8-bit RAM	25 ns (30 ns with 50% duty cycle)
64 x 8-bit RAM	35 ns
Byte-wide Shift Register	20 ns

LCA Family

XC4000

Demonstrates

Fast RAM operation

Introduction

The timing requirements of the XC4000 RAM primitive are that the address must be set up before the start of the Write Enable pulse (WE) and held for a short time after its end; data must be set up before the end of WE and held until its end. While such requirements are not unusual, they are not easily met in an LCA device.

Conventionally, the presence or absence of WE determines whether a RAM cycle is a read or write. Operating in this way, time is wasted generating the WE pulse each cycle. In larger RAMs, WE must also be gated to individual banks according to the address, wasting still more time. In addition, a clock rate twice the RAM-cycle rate is usually required to control the timing skew created in the generation and distribution WE.

With read-modify-write operation, the RAM is written every cycle. In write cycles, new data is written into the RAM; in read cycles, the data read from the RAM is rewritten, leaving the contents of the RAM unchanged. The unconditional write permits the clock signal to be used directly as WE, with WE asserted while the clock is High. The guaranteed low skew of the dedicated clock distribution nets simplifies the design, and increases its performance.

RAM Operation

A 16-word RAM is shown in Figure 1. The clock signal is used as the active-High WE, and no WE gating is provided. During the clock-Low period the address sets up on the RAM, data is read and registered on the rising edge of the clock. The flip-flops used for this register are available in every RAM-configured CLB and cannot be used independently since the DIN and H1 inputs are used in the RAM operation.

The Write signal determines whether old or new data is written during the clock-High period. The selected data must be set-up an appropriate time before the end of WE and held until the end of WE. This condition is easily satisfied if changes in the Input-Data and Write signals are triggered by the falling edge of Clock/WE.

The address hold time is satisfied by the output delay of the address register, provided that **both are driven directly from the same global clock net**. This can only be achieved using a BUFGS as the clock driver. The clock inverter is absorbed into the CLB to select the active clock edge, and does not create skew.

Several factors control the performance: The minimum clock-Low time is determined by the clock-to-set-up time from the address register to the read-data register, the minimum clock-High time is determined by the clock-to-set-up time from the read-data register to the trailing edge of WE in the RAM, and the total cycle time must permit both the Input Data and the Write signal to set up before the trailing edge of WE.

Figure 2 shows a 64-word RAM. Two banks of 32 words are used and both banks are written unconditionally. Write selection between the two banks is achieved using separate input data multiplexers. The Write controls is ANDed with bank select controls decoded from the address. Thus, data is only written to the appropriate bank. The output data from the appropriate bank is selected in a multiplexer to provide the read data output.

The pipeline stage at the multiplexer output is optional. If it is necessary to access the write data as it is being written, an additional connection may be made directly to the RAM outputs, and the data captured on the rising edge of the clock.

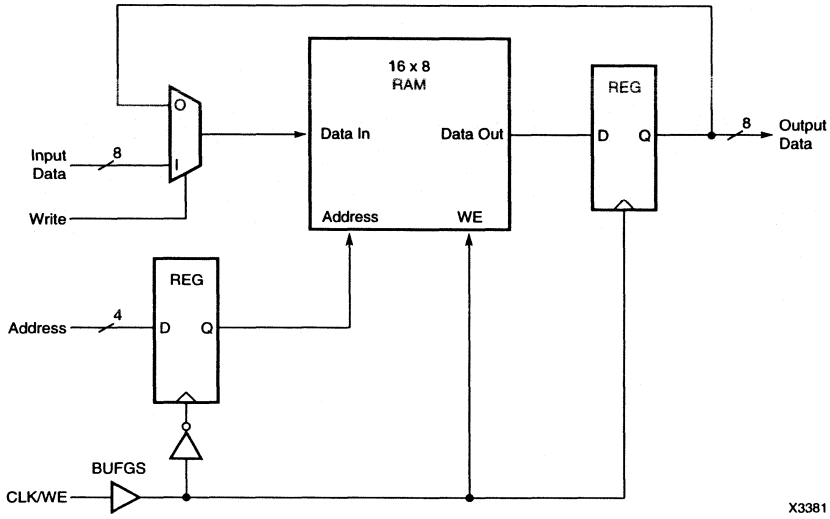


Figure 1. 16 x 8 RAM

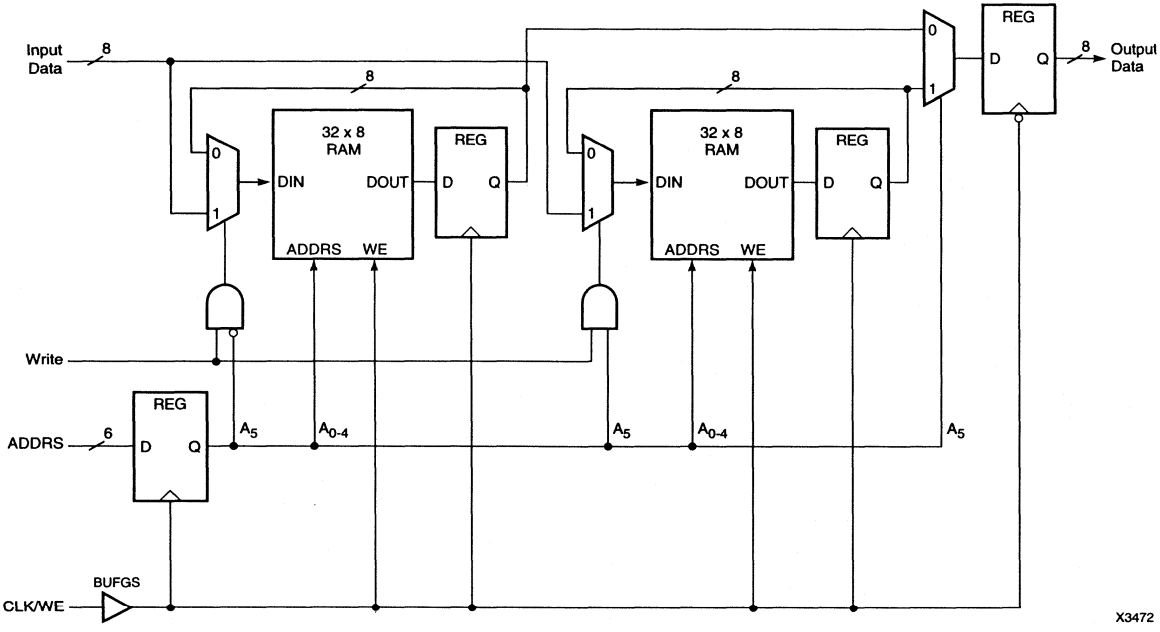


Figure 2. 64 x 8 RAM

Shift Register

Large shift registers can be implemented in RAM, as shown in Figure 3. In this design, shift-register segments of 16-words or less are concatenated to any required length. Sixteen is chosen to minimize the RAM cycle time, and eliminate the need for bank switching or multiplexing.

In Figure 3, each segment has its own address counter. This is unnecessary in most cases, since address counters with the same modulus can be shared. A typical shift register might have one modulo-16 address counter shared among all segments except the last. The last segment has a separate shorter address counter to provide the desired length. The shift register can be tapped between any two segments, and separate address counters can provide arbitrary taps.

The address counters need not be conventional binary counters. Provided the address sequence cycles repeatedly, the order is irrelevant. A 4-bit counter with any desired sequence requires only two CLBs; all four flip-flop are fed back to the four function generators, which determine the next state of their respective flip-flops.

The shift register in Figure 3 does not have a shift-enable control. A multiplexer to re-write existing data into the RAM is, therefore, unnecessary since a write is performed every cycle.

Shift-Enable control can be added, using the clock enable scheme shown in Figure 4. New data is written into the RAM on every clock cycle, even when not shifting. The RAM address, however, is not updated while shift is disabled. Instead, it is held constant, and the corresponding RAM location is over-written repeatedly. The only data retained in the RAM is the data entered as the shift is re-enabled permitting the address counter to advance.

When shifting is disabled, the data that is stored in the register at the RAM output remains there until shifting resumes. At this time. it is passed to the final output register or to the next RAM. The final output register prevents the output from changing when shift is first disabled. It does not alter the length of the shift register; with a 16-word RAM, data appears at the output as a result of the falling clock edge exactly 16 clocks after the falling edge on which the data was clocked in. Without the register, data appears one-half clock earlier, and when shift is disabled, would already be present.

No additional register is required between shift-register blocks. However, if there is a shift-register tap between blocks, the tap must be provided with an output register. The Shift-Enable control must be valid to enable or disable registers and counters clocked on the falling clock edge.

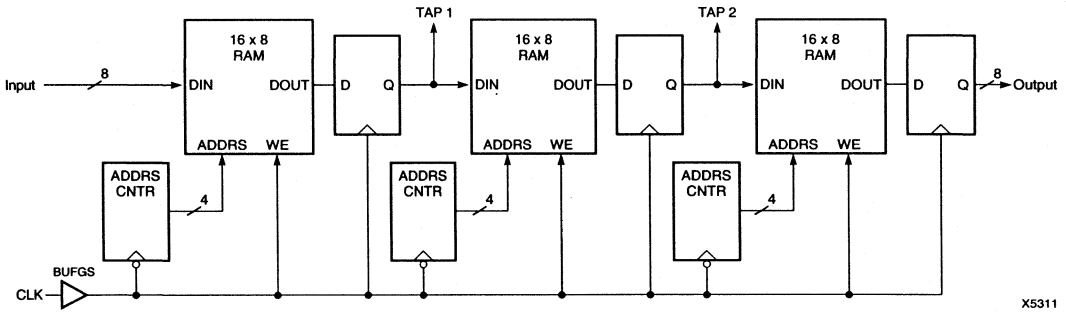


Figure 3. Byte-wide RAM-based Shift Register

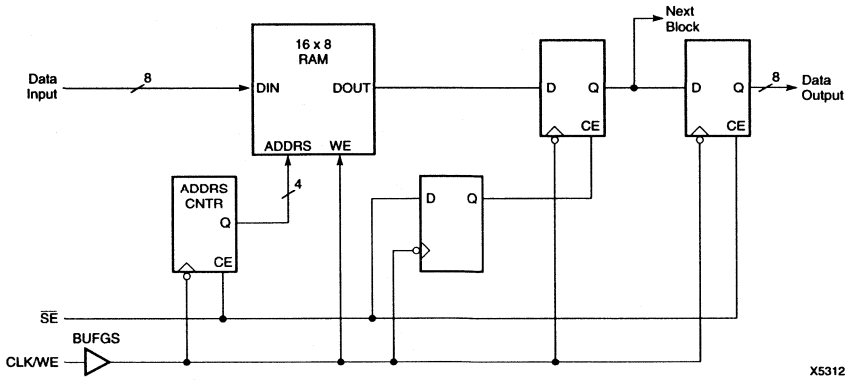


Figure 4. RAM-Based Shift Register with Shift Enable

High-Performance RAM-Based FIFO

XAPP 044.000

Application Note By BERNIE NEW

Summary

Two FIFO designs are described. In both cases, arbitration permits any RAM cycle to be a PUSH or a POP. XC4000 RAM performance is improved through read-modify-write operation, and the fastest clock required is at the RAM-cycle rate. The first design is expandable to any size FIFO, while the second, faster design is restricted to 16 or 32 words.

Specifications

Maximum Clock Frequency (estimated for XC4000-5)

16 x 8-bit FIFO 40 MHz

LCA Family

XC4000

Demonstrates

RAM Operation

Introduction

The four components of a RAM-based FIFO are shown in Figure 1. To control the RAM, the address-logic block maintains two addresses, one for the current write location, where data is PUSHed, and one for the current read location, from which data is POPped. Following a PUSH or a POP, the corresponding address is incremented, causing data to be written and read sequentially.

The flag logic uses the read and write addresses to determine the status of the memory. If the memory contains no valid data, an EMPTY flag is created; a FULL flag is created when all memory locations are occupied.

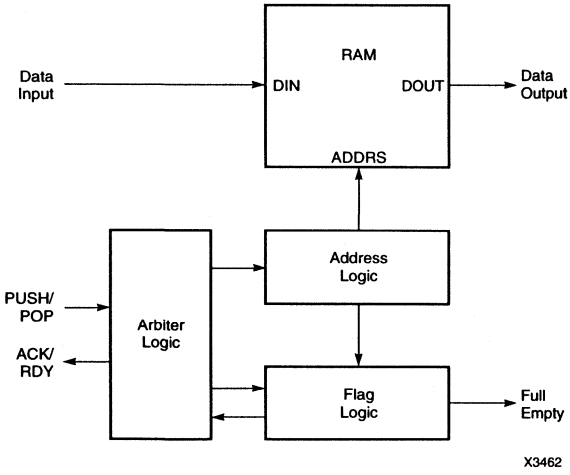


Figure 1. FIFO Block Diagram

The arbitration logic, for the most part, simply passes PUSH and POP requests to the RAM. Simultaneous PUSH and POP requests, however, must be resolved. The simplest schemes have a fixed priority, with either PUSH or POP being designated as the priority operation. If the priority operation is not possible because the RAM is full or empty, the priority should be overridden. Other schemes can alternate in priority between PUSH and POP, or favor one operation while its request persists.

Both FIFOs described depend on read-modify-write operation of the RAM, with Write Enable asserted every cycle. In the first design, a data multiplexer permits "non-write" cycles by rewriting existing data into the RAM. The same multiplexer provides bank selection for RAM expansion, permitting any size FIFO.

In the second design, the data multiplexer is omitted for faster operation. As a consequence, however, bank selection is not possible, and the RAM depth is limited to one CLB. Without a data multiplexer, only new data can be written in the RAM. During non-POP cycles, new data is always written to the current write address. If PUSH is asserted, the data is valid and the write address is incremented before the next write. If PUSH is not asserted, the address is not incremented, and the invalid data is subsequently overwritten. During a POP cycle, data read from the RAM is stored in a register, and the RAM location immediately overwritten with invalid data.

Operating Description

Expandable FIFO

Figure 2 shows the RAM and its address counters. A key element is the use of a BUFGS to drive the RAM Write Enable and clock the address register. The low skew of the global net ensures that the data and address hold

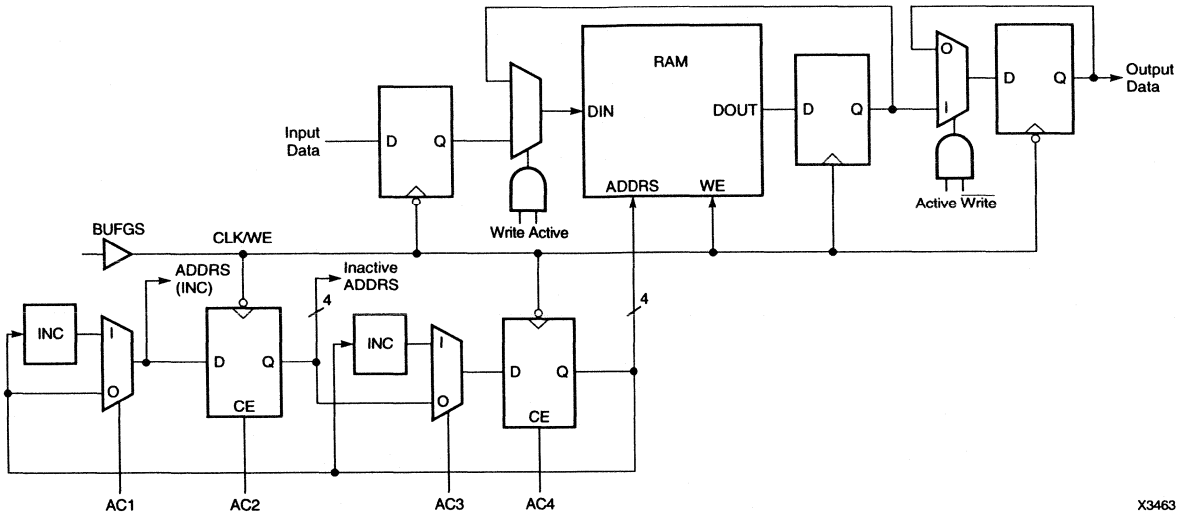


Figure 2. RAM and Address Logic

times are met. The RAM is written every clock cycle, and the multiplexer preceding the RAM determines whether new data is entered or the old data is re-entered. If it is necessary to expand the RAM, a bank select signal, derived from the address counter, can be ANDed with the Active signal to limit writing to a single bank of RAM. In addition, a read-data-select multiplexer must be provided.

The address-generation logic is shown in the lower portion of Figure 2. The right-hand register contains the address currently being used by the RAM, the write address during a PUSH and the read address during a POP. The left-hand register contains the address that is not being used.

The address-logic instruction set, Table 1, permits the active address to be incremented and remain active for successive PUSHes or POPs, or be incremented and become inactive, for a PUSH followed by a POP, or vice versa. It also permits the addresses to remain unchanged or simply interchanged.

$\overline{\text{FULL}}$ and $\overline{\text{EMPTY}}$ flags are generated in the flag logic, Figure 3. Flags can only be asserted or de-asserted during active PUSH or POP cycles, and both are triggered by the incremented active RAM address becoming equal to the inactive address. If this equality occurs during a PUSH cycle, the new write address contains the next data to be POPed, and the FIFO is full. If equality is reached during a POP, the FIFO is empty since the new read address is waiting to be written in the next PUSH operation.

Table 1. Address Logic Instruction Set

Current Op R/W	Act/Inact	Next Op R/W	AC 1 2 3 4
R	Active	R	1 0 1 1
R	Inactive	R	X 0 X 0
W	Active	R	1 1 0 1
W	Inactive	R	0 1 0 1
R	Active	W	1 1 0 1
R	Inactive	W	0 1 0 1
W	Active	W	1 0 1 1
W	Inactive	W	X 0 X 0

X3464

Consequently, the flags can be generated by gating the comparator output with the Write signal and registering it during active RAM cycles. The address-logic instruction set is constructed such that the inactive address and the incremented active address are always available to the comparator. The flags clear on the next active RAM cycle, when the addresses become non-equal. For correct operation, this cycle must be a PUSH if the FIFO is empty or a POP if it is full.

Figure 4 shows a simple arbitration circuit. The PUSH and POP inputs control a multiplexer that determines the operation in the next RAM cycle. If PUSH only is asserted, the next cycle is a write. If POP only is asserted, it is a read

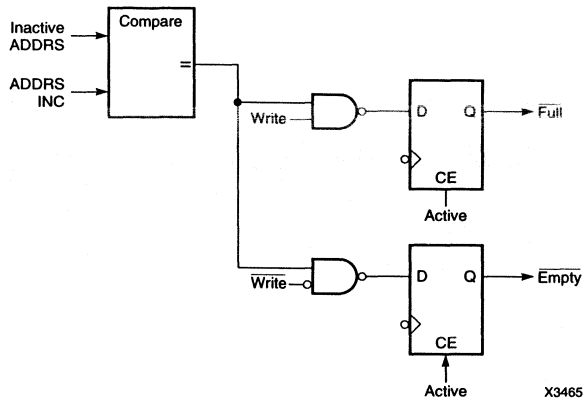


Figure 3. Flag Logic

cycle. If both are asserted together, the next operation in the next RAM cycle is determined by a user-defined Priority signal. Several options for the Priority signal are discussed later.

If a PUSH or a POP is requested and the FIFO is not full or empty, respectively, the next cycle is declared Active. A write or a read occurs and the corresponding address is incremented. Otherwise, the cycle is inactive; no read or write occurs and the addresses remain unchanged. In an inactive cycle, the Write signal is de-asserted by default.

Two handshake signals are generated. ACK acknowledges that a PUSH request will be honored in the next RAM cycle. Input data is captured on the falling clock edge that starts the RAM cycle. RDY indicates that a POP request will be honored during the next RAM cycle. Output data is made available on the falling clock edge that ends the RAM cycle.

In deciding the next operation when both PUSH and POP are asserted, the most straightforward Priority functions simply default to one operation or the other. To always write, a logic High could be used, and to always read, a logic Low. In practice, however, this can lead to wasted cycle. For example, PUSH could win when the FIFO is full and the operation cannot be performed. A better choice is to use $\overline{\text{FULL}}$ as Priority to always select write unless the FIFO is full. Similarly, using $\overline{\text{EMPTY}}$ will cause POP to always win unless the FIFO is empty.

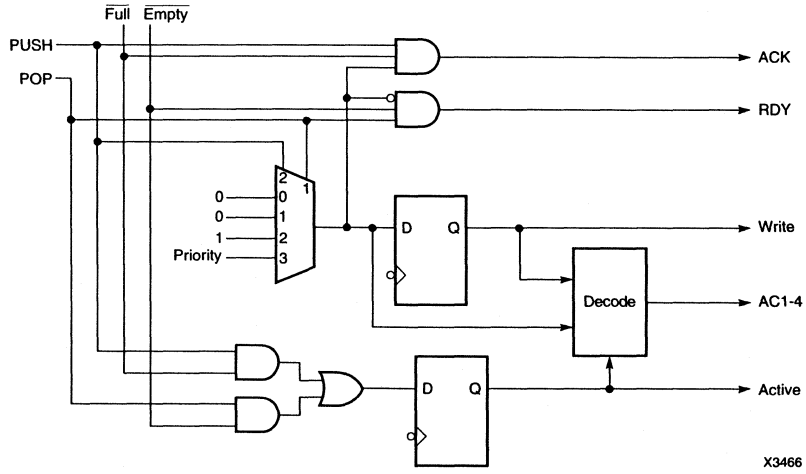


Figure 4. PUSH/POP Arbitration Logic

The above priorities are useful when receiving data from a burst source, such as a bus, or transmitting burst data. While burst is in progress, however, the other operation can be locked out for many cycles. If a more even resource allocation is required, $\overline{\text{Write}}$ can be used as Priority. In this case, requesting PUSH and POP continuously results in alternating reads and writes.

While the time to acquire the FIFO is reduced to no more than one cycle, the guaranteed peak PUSH/ POP rate is also reduced. In the limit, PUSH and POP may only operate at half the RAM cycle rate. The average data through the FIFO is unaffected, however. In the long term, it is obvious that no more than half the RAM cycles can be PUSHes. Attempting to achieve more will fail when the FIFO becomes full. Similarly, no more than half the cycles can be POPs, since the FIFO will become empty.

A third option permits both burst reads and burst writes, although either PUSH or POP may experience a long delay acquiring the FIFO if it is busy. Priority is connected to Write. As a result, the FIFO repeats its last operation whenever there is a conflict. A burst read or write will continue, and lock out the other operation until the burst is complete.

High-Performance FIFO

The RAM block for the high-performance FIFO is shown in Figure 5. In this design, the RAM has simple input and output registers. The input register captures data on the falling edge of the clock which, in addition, marks the start of a RAM cycle. Data is captured in the output register at the end of the read phase, when the clock goes high.

Prior to the start of the RAM cycle, a selection is made between the read and write addresses, and the selected address is registered when the RAM cycle starts. The read address is only selected when a POP is to be executed. Stable output data is retained from POP to POP, however, since the output register is only enabled during POPs. If the output data is required to change on the falling edge, an additional register must be used.

Since new data is written into the RAM every cycle, the read address cannot be selected during idle cycles; valid data waiting to be read would be destroyed by the write. Consequently, the write address is selected for both PUSH and idle cycles.

In the previous design, the RAM can be filled completely. When the FIFO is full, the next write address becomes equal to the next read address. This is not a problem, provided the location is read before it is overwritten. In the current design, there must always remain at least one unused location where invalid data is written during idle cycles.

When a PUSH occurs, the data written finally becomes valid, and the write address is incremented. During subsequent idle cycles, invalid data is written to the new write address. Overwriting of this address continues until the next PUSH.

As a consequence, a maximum of 15 words can be stored in the RAM. The FIFO can, however, store two more words in the input and output registers. The total storage capacity is 17 words.

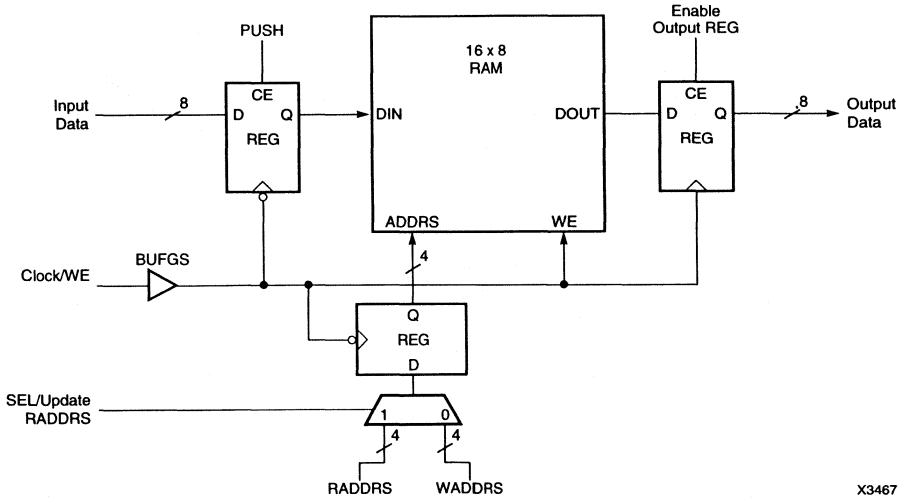


Figure 5. RAM Diagram

Figure 6 shows the address and flag logic for the FIFO. Modified 4-bit linear-feedback-shift-register (LFSR) counters are used. Conventionally, 4-bit LFSR counter use the XNOR of the last two shift-register bits as feedback the input. This results in a sequence that repeats every 15 clocks. The missing count, all-1s, can, however, be added to provide the count sequence shown in Table 2.

Normally, 1110 is followed by 0111, and 1111 would cause the counter to lock up. To include 1111 in the sequence, 111X is detected, and the shift-register input is inverted while this condition is met. Consequently, 1110 is followed by 1111 and the next count is 0111, since the input remains inverted. The remainder of the count sequence is unaffected.

A benefit of LFSR counters in this design is that adding one extra bit to the shift register permits access to two adjacent addresses, which, in turn, permits easy generation of the $\overline{\text{FULL}}$ flag. The current write address is available to the RAM, while the next write address is also available for comparison with the read address. When the next write address equals the read address the FIFO is full. The current write address is the sixteenth RAM location needed for invalid data writes during idle cycles. The FIFO is empty when the current read address becomes equal to the current write address, which is yet to be written with valid data.

Table 2. Adder-Counter Sequence

Shift Register Input →
0 0 0 0
1 0 0 0
1 1 0 0
1 1 1 0
1 1 1 1
0 1 1 1
1 0 1 1
1 1 0 1
0 1 1 0
0 0 1 1
1 0 0 1
0 1 0 0
1 0 1 0
0 1 0 1
0 0 1 0
0 0 0 1
0 0 0 0
1 0 0 0
⋮

X5281

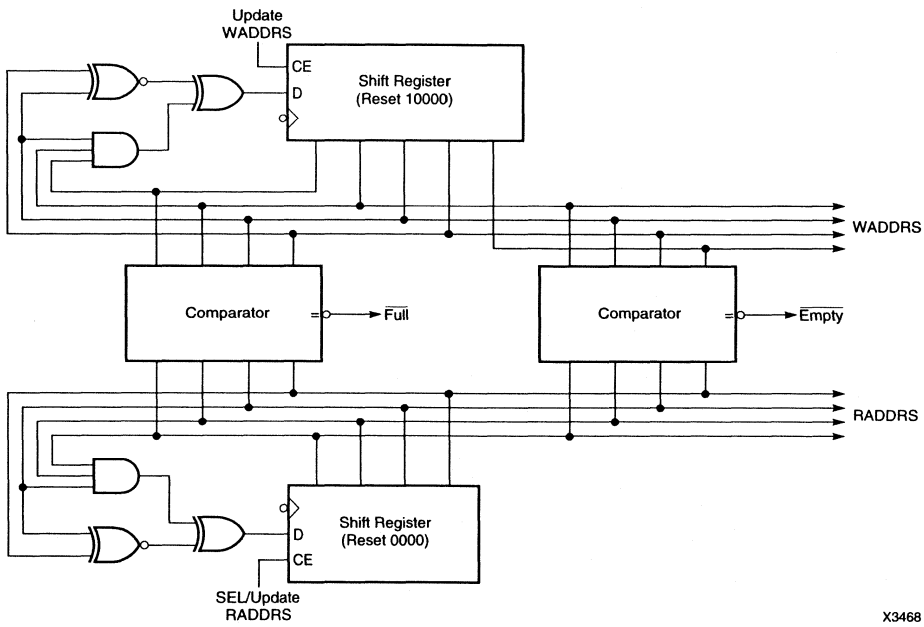


Figure 6. Address/Flag Logic

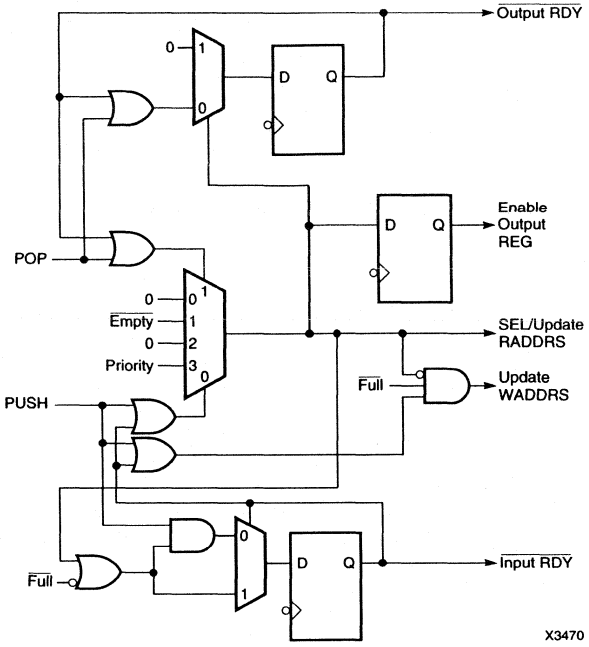


Figure 7. PUSH/POP Arbitration

The arbitration logic is shown in Figure 7. As in the previous design, the core of the arbiter is a multiplexer that selects the next operation according to the requests and the status of the RAM. The Priority input must be defined and implemented as discussed previously.

The output of the multiplexer is High when a read is to be performed. The Highfi causes the read address to be both selected for the RAM and simultaneously updated. Otherwise, the write address is selected. The write address is only updated, however, when a PUSH is requested and the UPDATE-WADDRS command is issued.

PUSH/POP requests and input data must set up to the falling edge of the clock, and POPed data becomes available on the subsequent rising edge. If a request cannot be serviced immediately, it is stored in one of two flip-flops, and a RDY is asserted on the falling clock edge at the start of the RAM cycle. If a request can be serviced, the corresponding $\overline{RDY}$ flag is never asserted.

When a PUSH is deferred, the input data is still captured in the input register, but it is not transferred to the RAM. In this case, $\overline{RDY}$ should suppress further PUSHes. The $\overline{RDY}$ flags are cleared at the start of the RAM cycle in which the request is serviced.

Megabit FIFO in Two Chips: One LCA Device and One DRAM

XAPP 030.000

Application Note By PETER ALFKE

Summary

This Application Note describes the use of an LCA device as an address controller that permits a standard DRAM to be used as deep FIFO.

Xilinx Family

XC3000/XC3100

Demonstrates

Non-linear Counters
Pseudo-random RAM Addressing

Introduction

A bit-serial FIFO buffer is a general-purpose tool to relieve system bottlenecks, e.g., in LANs, in communications, and in the interface between computers and peripherals. Small FIFOs are usually designed as asynchronous shift registers, but a larger FIFO with more than 256 locations is better implemented as a controller plus a two-port RAM, or as a controller plus a single-port RAM, either SRAM or DRAM.

SRAMs are fast and easy to use, but at least four times more expensive than DRAMs of equivalent size. Dynamic RAMs offer lower-cost data storage, but require complex timing and address multiplexing, which makes them unattractive in small designs. For FIFOs with more than 256K capacity, a DRAM offers the lowest cost solution, if the controller can be implemented in a compact and cost-effective way. An XC3020 Logic Cell Array can easily perform all the control and addressing functions with many gates left over for additional features.The XC3020 can be programmed to control one or more DRAMs for a FIFO of up to 16 megabytes, with data rates up to 16 Mbits per second serially or 16 Mbytes per second byte-parallel.

Logic Description

This FIFO DRAM controller comprises the following.

- Input/output buffer with synchronizing logic
- 20-bit Write pointer (counter)
- 20-bit Read pointer (counter)
- 20-bit full/empty comparator
- 10-bit address multiplexer
- Control and arbitration logic

Figure 1 is a block diagram of the FIFO Controller. The Write pointer defines the memory location where the incoming data is to be written, while the Read pointer defines the memory location where the next data can be read. The identity comparator between the address pointers signals when the FIFO is full or empty.

When the Write and Read pointers become identical as a result of a Write operation, the FIFO is full, and further Write operations must be prevented until data has been read out to create space in the memory. If the two pointers become identical as a result of a Read operation, the FIFO is empty and further Read operations must be prevented until new data has been written in. With a single-port RAM, Read and Write operations must be inherently sequential, and there is no danger of confusing the full and empty state, a problem that has plagued some two-port designs.

A straightforward design would use synchronous binary counters for the two pointers, but it is far more efficient to use linear feedback shift-register (LFSR) counters. Such counters require significantly less logic and are faster since they avoid the carry propagation delay inherent in binary counters. LFSR counters have two peculiarities: they count in a pseudo-random sequence, and they usually skip one state, i.e., a 20-bit LFSR counter repeats after 2^{20}-1 clock pulses. In a FIFO Controller, both these issues are irrelevant; the address sequence is arbitrary, provided both counters sequence identically.

The RAS/CAS multiplexing of the 20-bit address is performed without an explicit multiplexer. Every other bit of the shift-register counter is used to provide the 10-bit address. Before the incrementing shift, these bits are used as the Row address. After incrementing, they are used as the Column address. The Column address of any position is thus identical with the Row address of the following position, but since the binary sequence of a shift register counter is pseudo-random anyhow, this is not a problem.

The address generation logic is shown in Figure 2. With this design, two shift-register counter bits fit into one XC3000-series CLB, with the identity comparator using the combinatorial portion of the same CLB, Figure 3.

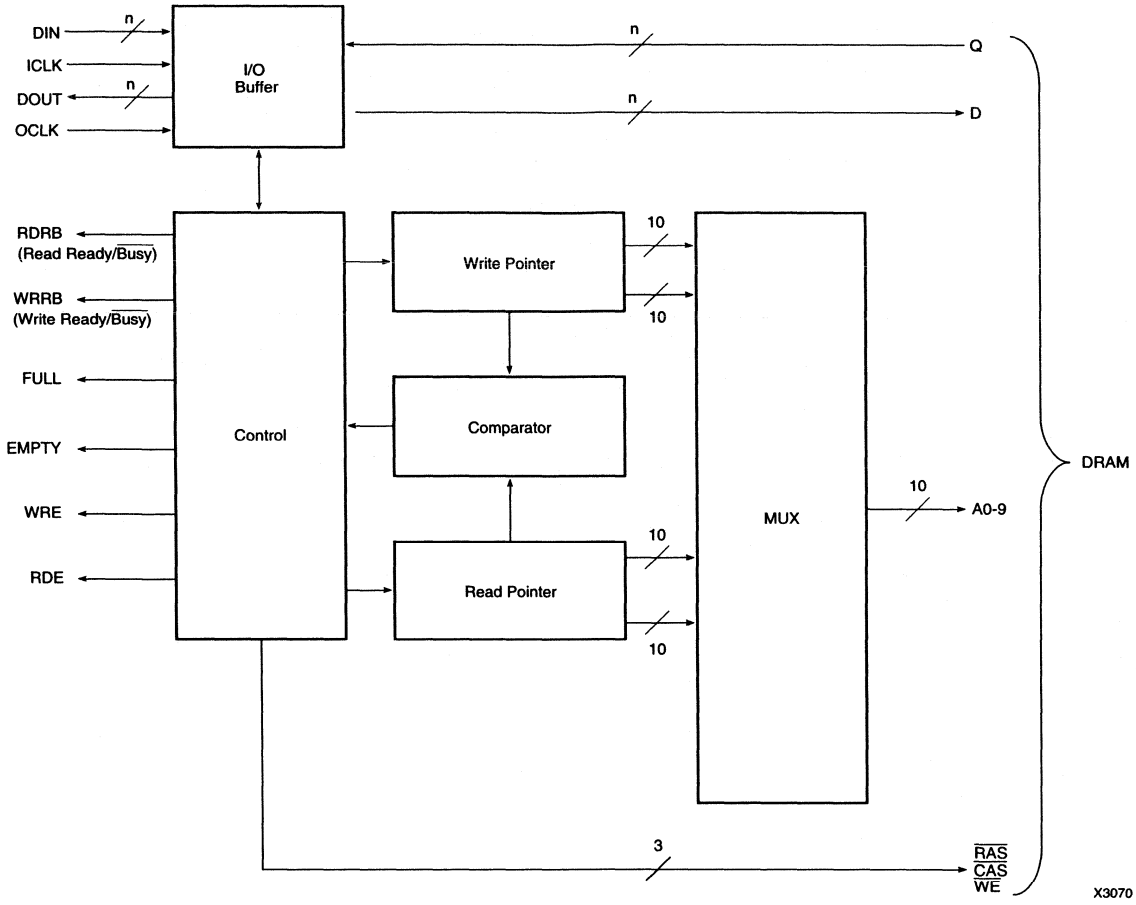


Figure 1. Megabit FIFO Controller in an XC3020

The FIFO controller permits the user to perform totally asynchronous Read and Write operations, while it synchronizes communication with the DRAM. The design takes advantage of the DRAM internal refresh counter by using CAS-before-RAS refresh/address strobes.

Both 20-bit pointers, plus their 20-bit identity comparator, plus the Row/Column multiplexer thus fit into only 20 CLBs; refresh timer and address multiplexer use another 10 CLBs and the data buffer plus control and arbitration logic take another 23 CLBs, for a total of 53, an easy fit in an XC3020.

This design can easily be modified for larger or smaller DRAMs. Other variations that might be considered are: multiple parallel bits, e.g., byte-parallel operation, interrupt-driven control, multiplexed data for multiple parallel-bit storage, and byte parallel storage with bit-serial I/O. This latter case requires special attention when the FIFO is emptied after a non-integer number of bytes has been entered, and requires direct communication between the input Serial-to-Parallel converter and the output Parallel-to-Serial converter.

This design is available from Xilinx. Call the Applications Hot Line 408-559-7778 or 1-800-255-7778.

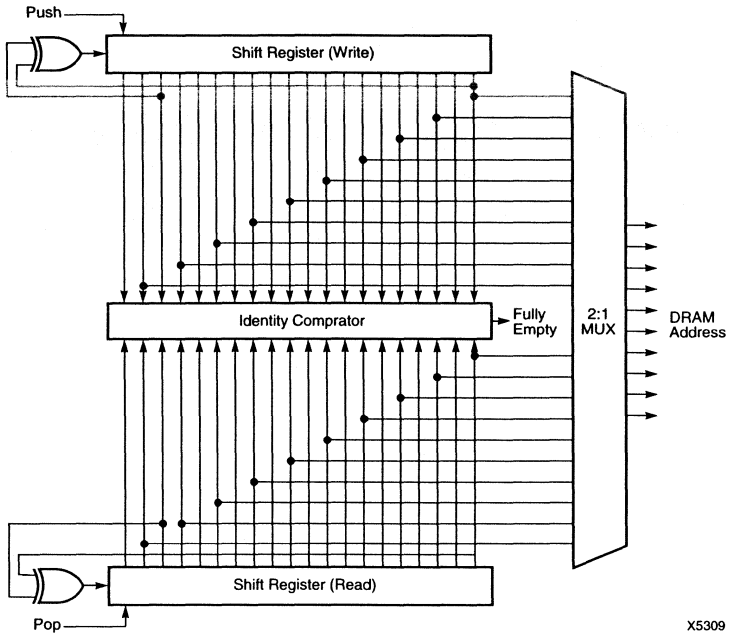


Figure 2. DRAM Address Generation

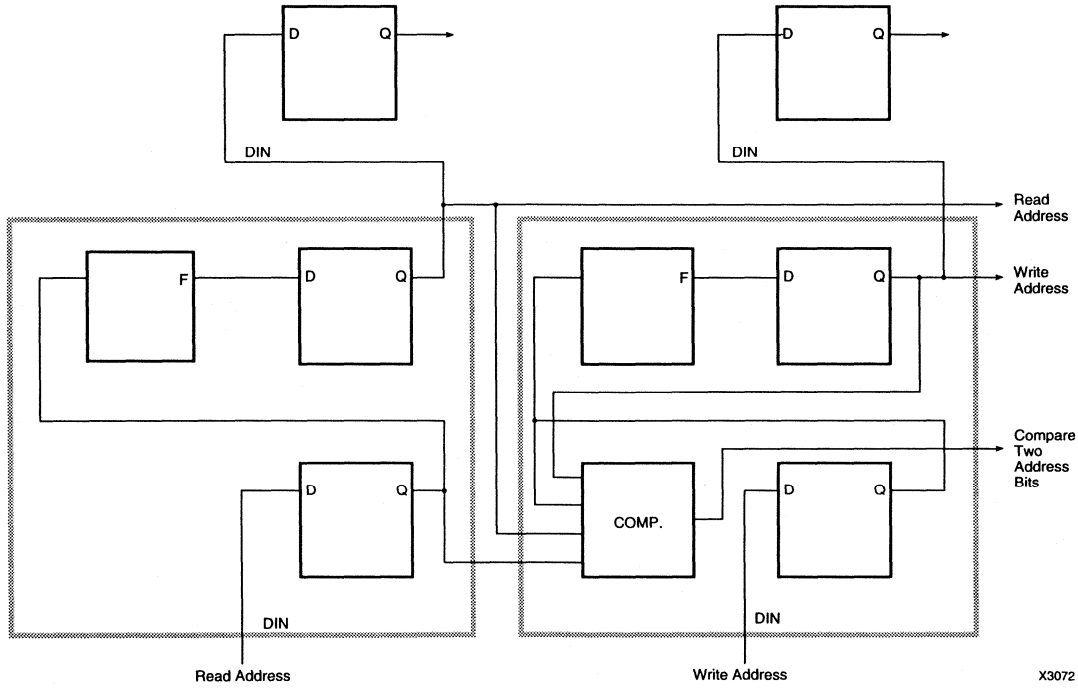


Figure 3. 2-Bit Slice of Two Counters and Comparator in Two CLBs

Digital Mixer in an XC7272

XAPP 035.002

Application Note By DAN UJVARI Applied Technical Marketing

Summary

This Application Note describes a simple mixer that operates at video rates, and provides 9 levels of mixing.

Xilinx Family

XC7200/XC7300

Demonstrates

High-speed Arithmetic

Introduction

A digital mixer provides for controlled transition from one incoming digitized analog data source to another. A typical application is in broadcast television where the switching between picture sources should be gradual. The XC7272 can implement such a digital mixer running at a 40-MHz sample rate. It handles two incoming 8-bit data streams, A and B, and mixes them in nine steps, controlled by a 4-bit coefficient, N.

$$\text{Output} = A\frac{N}{8} - B\frac{N}{8}$$

where N = 0, 1, 2....8

Operation

The design consists of two processing channels, A and B, combined in an output adder, Figure 1. The two channels are identical in structure, but are driven by complementary coefficients. Each 8-bit data stream is multiplied by its coefficient by adding or subtracting the outputs of two shift arrays, Shift1 and Shift2.

Shift1 can shift by 0,1, 2, or 3 positions, and can disable its output. Thus, it can multiply the data by 1, 1/2, 1/4, 1/8, or 0. Shift2 can shift by 2 or 3 positions and can also disable its output, thus multiplying the data by 1/4, 1/8, or 0. Table 1 below describes the operation of the complete mixer.

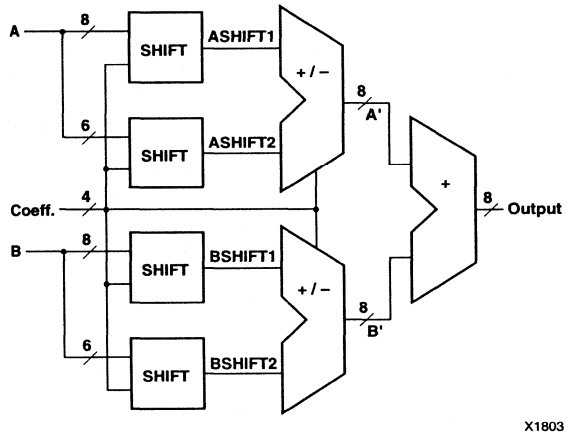


Figure 1. 40 MHz Digital Mixer

The design is pipelined for fastest throughput, and has been implemented in one XC7272A. It uses 56 of the available 72 Macrocells and 28 signal pins: 16 data inputs, four coefficient inputs, eight data outputs and one clock input. There is room for enhancements such as larger incoming word length or finer coefficient granularity.

Table 1: Mixer Operation

Coeff	ASHFT1	ASHFT2	A'	BSHFT1	BSHFT2	B'	Output
0	0	0	0	B	0	B	B
1	0.125A	0	0.125A	B	0.125B	0.875B	0.125A + 0.875B
2	0.250A	0	0.250A	0.50B	0.250B	0.750B	0.250A + 0.750B
3	0.250A	0.125A	0.375A	0.50B	0.125B	0.625B	0.375A + 0.625B
4	0.500A	0	0.500A	0.50B	0	0.500B	0.500A + 0.500B
5	0.500A	0.125A	0.625A	0.250B	0.125B	0.375B	0.625A + 0.375B
6	0.500A	0.250A	0.750A	0.250B	0	0.250B	0.750A + 0.250B
7	A	0.125A	0.875A	0.125B	0	0.125B	0.875A + 0.125B
8	A	0	A	0	0	0	A

Multiplexers and Barrel Shifters in XC3000/XC3100

XAPP 026.001 Application Note By PETER ALFKE and BERNIE NEW

Summary

This Application Note provides guidance for implementing high performance multiplexers and barrel shifters in XC3000 LCA devices.

Xilinx Family

XC3000/XC3100

Introduction

Since the function generator in the XC3000 series CLB has only five inputs, it cannot directly implement a four-input multiplexer, which requires four data inputs and two select inputs. The CLB does, however, have the logic capability to implement a 4-input multiplexer.

This applications shows how to access the full logic capability of the CLB for 4-input multiplexers. It also shows how best to implement larger multiplexers and barrel shifters.

Multiplexers

Four-Input Multiplexer

CLB function generators have a base-FGM operating mode that permits certain functions of more than five variables to be implemented. The restriction on the function is that it must be implementable as a multiplexer selecting between two functions, each of four variables. Clearly, a 4-input multiplexer meets this requirement; each 4-input function implements a 2-input multiplexer, and the final multiplexer selects one of the outputs.

Since the CLB only has five logic inputs to the function generators, the sixth input to the multiplexer must reach the function generators via the CLB .di pin, a flip-flop and the internal feedback path. Routing through a flip-flop has obvious timing implications, but using this path can result in through delay and resource savings of 50%. Often the additional select delay can easily be accommodated, and sometimes it even saves storage resources elsewhere.

One approach is to pipeline the select lines, Figure 1. Two bits of the 4-input multiplexer are implemented in two CLBs. In one CLB, the S_0 select line is registered, while in the other the S_1 select line is registered. In addition to being used within the CLB, the registered versions are output for use in the other CLB. This balances the delay in the select lines. Notice that the order of the multiplexer ranks is reversed in the two CLBs.

Alternatively, if the design requires one of the multiplexer inputs to be pipelined, this input may use the flip-flop route, thus saving an external pipeline register, Figure 2. In either case, one CLB flip-flop remains available for optional use registering the multiplexer output.

Wider Multiplexers

If the multiplexer select line can be pipelined, large multiplexers are best implemented using multiple ranks of the 4-input multiplexer described above, together with a 2-input multiplexer, if required. Even if a completely combinatorial circuit is absolutely necessary, there are better alternatives to using multiple ranks of 2-input multiplexers.

While 4-input multiplexers cannot be implemented in a single CLB, it is possible to implement a 3-input multiplexer in one CLB. If this 3-input multiplexer is considered part of a 4-input multiplexer that is completed elsewhere, it can be used in expansion schemes, and binary encoding of the select lines can be retained.

The 8-input multiplexer, Figure 3, uses two 3-input multiplexers and a 2-input multiplexer to select one bit from six; on the two outstanding select codes, Zeroes are selected. These two select codes are also used to AND the corresponding inputs into a 2-input multiplexer. The output of this multiplexer is Zero whenever one of the other six select codes is asserted, and consequently, it is only necessary to OR the two outputs to complete the multiplexer.

This structure requires four CLBs, as does the 2-input multiplexer approach. However, the delay is only two CLBs instead of three, a reduction of 33%.

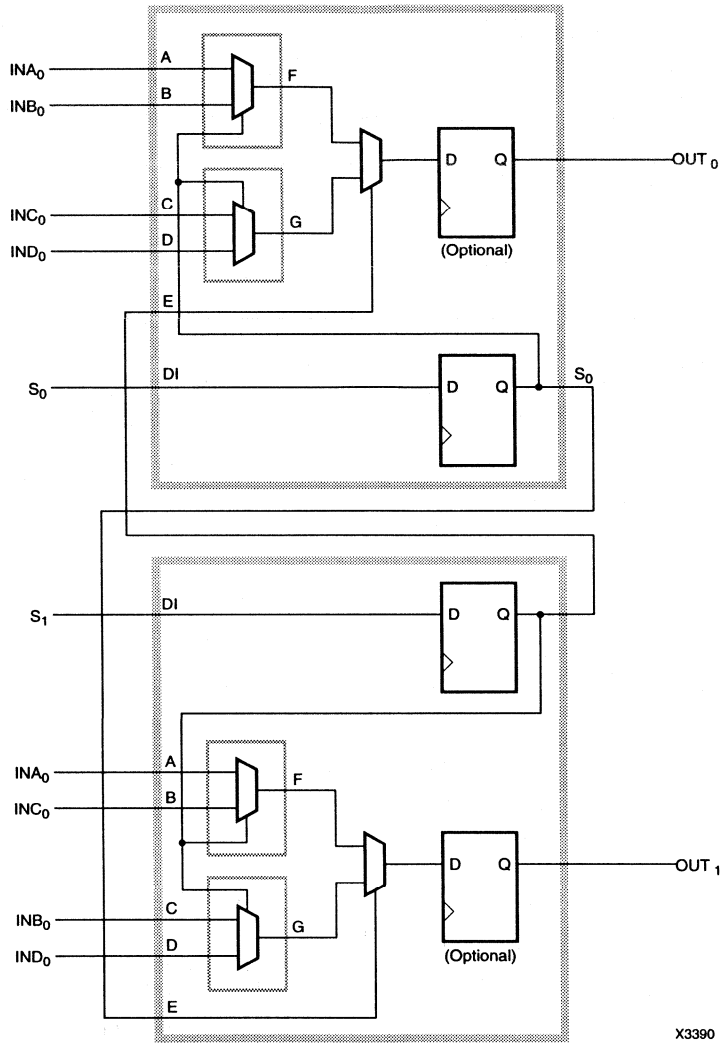


Figure 1. Dual 4:1 Multiplexer with Pipelined Select (Two CLBs)

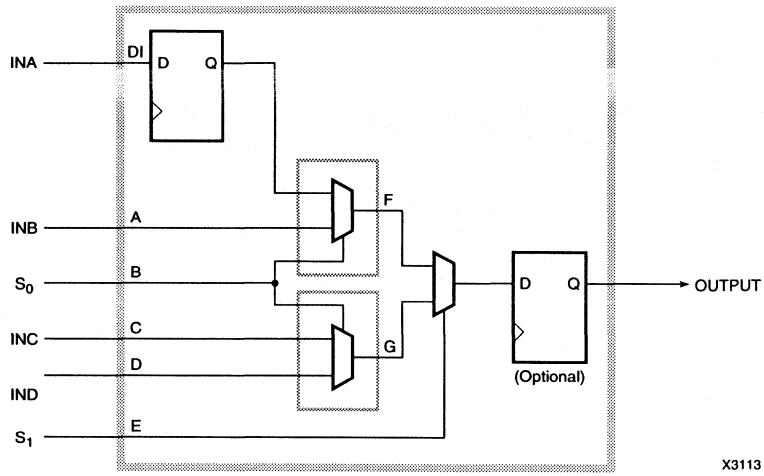


Figure 2. 4:1 Multiplexer with Pipelined Input

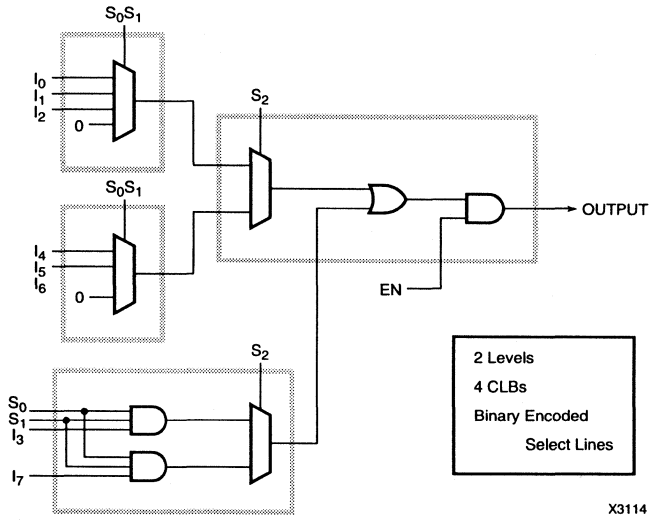


Figure 3. 8:1 Multiplexer

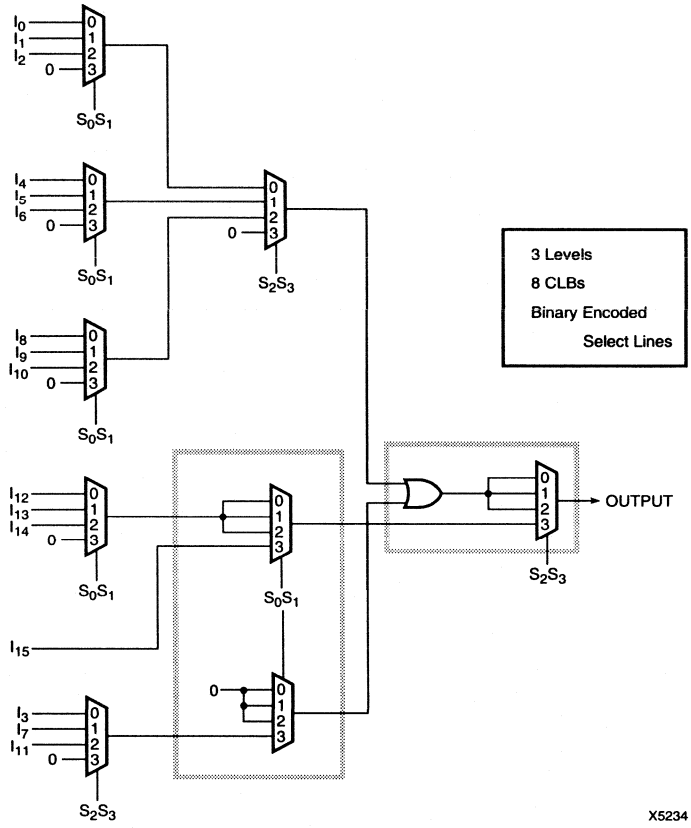


Figure 4. 16:1 Multiplexer

An output enable control is provided that permits the multiplexer to be expanded by ORing the outputs in an additional level of logic. A single CLB can implement a 5-input OR gate. Consequently, this expansion scheme can accommodate up to 40-input multiplexers within three levels of CLBs. The more significant select lines must be decoded to provided individual enables to each 8-input multiplexer, but this logic settles in parallel with the first level of CLBs.

For 16-input multiplexers, the design shown in Figure 4 may be used. It requires eight CLBs in three levels, which is one CLB fewer than is needed to combine two 8-input multiplexers, and one less level of CLB than a design based on 2-input multiplexers.

Barrel Shifters

A four-input barrel shifter has four data inputs, four data outputs and two control inputs that specify rotation by 0, 1, 2 or 3 positions. A simple approach would use four 4-input multiplexers, since each output can receive data from any input. This approach yields the best solution only if the select lines can be pipelined, and the 4-input multiplexer design described above is used. The complete barrel shifter can be implemented in one level of four CLBs.

If the barrel shifter must be fully combinatorial, it is better to decompose the barrel shifter into 2-stages, Figure 5. The first stage rotates the data by 0 or 1 positions, and

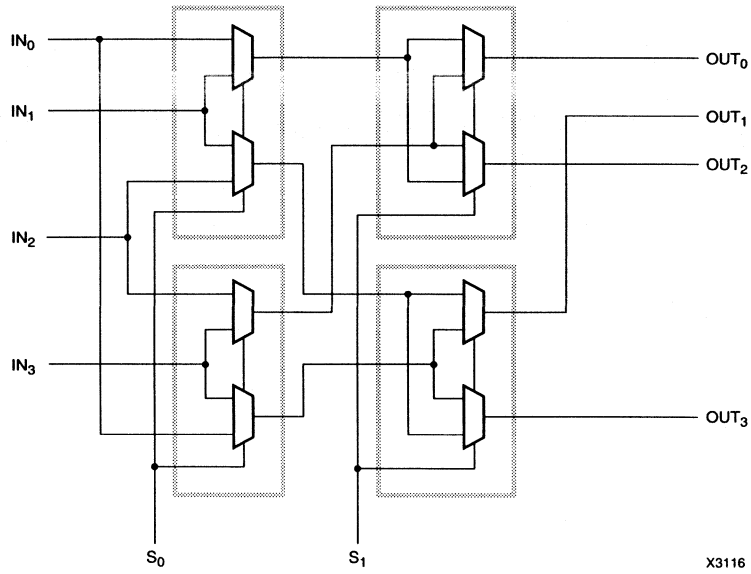


Figure 5. 4-Bit Barrel Shifter

the second rotates the result by 0 or 2 positions. Together, these two shifters provide the desired rotations of 0, 1, 2 or 3 positions. As in the previous design, four CLBs are required, but the number of levels increases to two. A combinatorial 4-input multiplexer approach would have used six CLBs in two levels.

This binary decomposition scheme can be used for any number of bits. The number of levels required for an N-bit shifter is $\log_2 N$, rounded to the next higher number if N is not a power of two. Each level requires $^N/_2$ CLBs. The first level rotates 0 or 1 positions, and subsequent levels each rotate by twice as many positions as the preceding level. The select bits to each level form a binary-encoded shift control.

For example, an 8-bit barrel shifter can be implemented in three levels of 2-input multiplexers that rotate by 1, 2 and 4 positions. Each level requires four CLBs, for a total of 12. For a 12-input barrel shifter, four levels of multiplexer are required. These multiplexers rotate by 1, 2, 4 and 8 positions, and require a total of 24 CLBs.

The 16-bit barrel shifter shown in Figure 6 has only two levels of CLB, and is, therefore, twice as fast as one using the 2-input multiplexer approach. However, the shift control must be pipelined, since it uses the 4-input multiplexer shown in Figure 1. The first level of multiplexers rotates by 0, 1, 2 or 3 positions, and the second by 0, 4, 8 or 12 positions. Each level requires 16 CLBs, and the total of 32 is the same as for the 2-input approach. The shift control remains binary.

Again, this scheme can be expanded to any number of bits using $\log_4 N$ rotators that successively rotate by four times as many bit positions. For sizes that are odd powers of two, the final level should consist of less costly 2-input multiplexers.

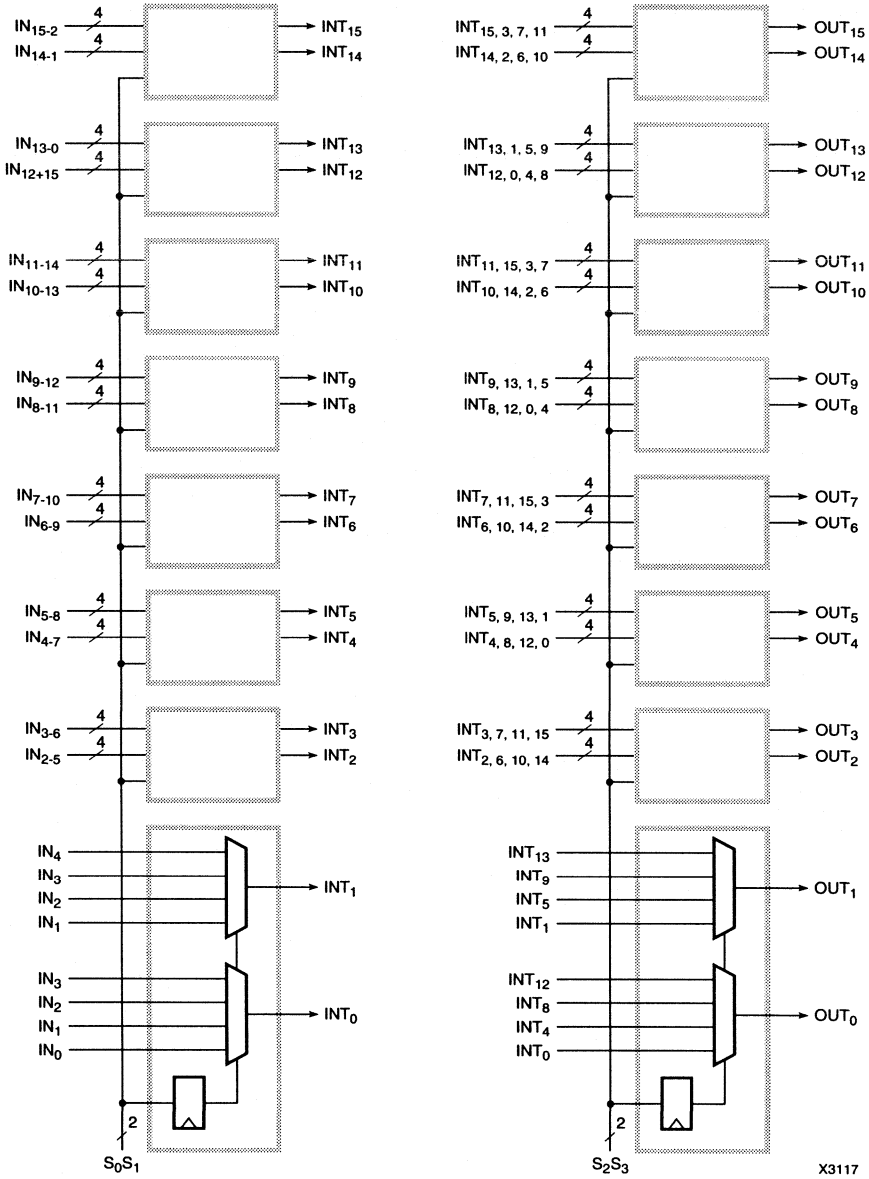


Figure 6. 16-Bit Barrel Shifter

Serial Code Conversion between BCD and Binary

XAPP 029.000 Application Note By PETER ALFKE and BERNIE NEW

Summary

Binary-to-BCD and BCD-to-binary conversions are performed between serial binary values and parallel BCD values.

Xilinx Family	*Demonstrates*
XC3000	Serial Arithmetic

Introduction

The LCA architecture with its powerful function generators evenly interspersed between flip-flops lends itself very well to serial code conversion. Data is entered into a register in one format, and retrieved from the same register in a different format. A common application of this technique is converting binary data to BCD, and BCD to binary.

Operating Description

Binary-to-BCD Conversion

Binary-to-BCD conversion is performed in a modified shift register that successively doubles its BCD contents. As shown in Figure 1, the binary data is shifted into the converter serially, MSB first. Subsequent bits are entered into the shift register to fill the LSB vacated by the doubling. The conversion is complete when all bits of the binary input have been entered, at which time the BCD result is available in parallel form. Each input bit will have been doubled and redoubled to regain its original binary weight, but in BCD format.

To remain a valid BCD number when doubled, a BCD digit of 5 or greater must not just be shifted, but must be converted into the proper BCD representation of its doubled value; along with a 1 being shifted into the next higher digit, a 5 is converted into a 0, a 6 into a 2, a 7 into a 4, an 8 into a 6, and a 9 into an 8.

The binary-to-BCD converter requires three CLBs for each BCD digit in the output, Figure 2. To start a new conversion, $\overline{\text{INIT}}$ should be asserted at the time the binary MSB is applied to the converter input. $\overline{\text{INIT}}$ clears all bits except the LSB which is loaded.

BCD-to-Binary Conversion

BCD-to-binary conversion reverses the process described above, Figure 3. BCD data is parallel loaded into a modified shift register that successively halves its contents. The equivalent binary value is obtained serially, LSB first, from the LSB of the shift register.

To divide by 2, data in the shift register is shifted towards the LSB. However, when a bit shifts across a digit boundary, its weight in the lower digit is 5. This value is added to the shifted digit using carry-save adders associated with bits 0 and 2. The conversion is complete when all bits of the binary output have been generated.

The BCD-to-binary converter requires three CLBs per digit, Figure 4. A new conversion is started by applying the BCD data and asserting the $\overline{\text{LD}}$ control to load the data. The MSB of each digit is loaded into the carry flip-flop of the bit-2 adder; the carry of the bit-0 adder is cleared.

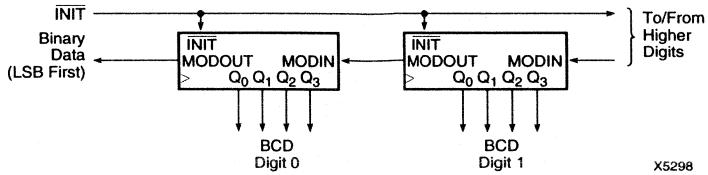


Figure 1. Binary-to-BCD Converter

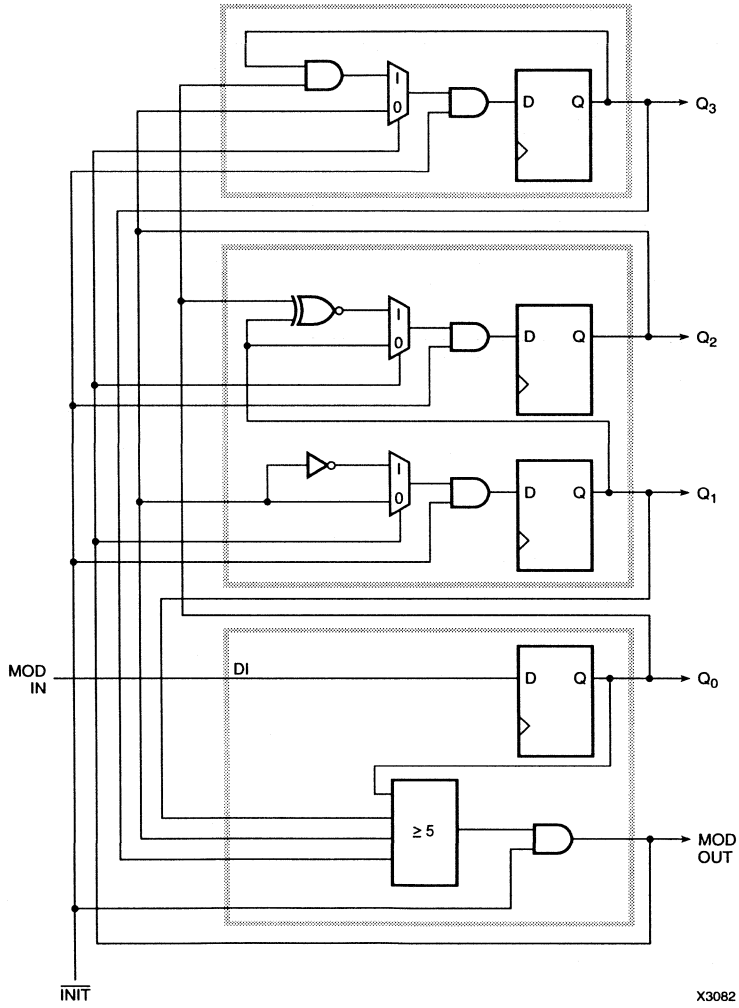


Figure 2. Binary-to-BCD Converter (Three CLBs per BCD Digit)

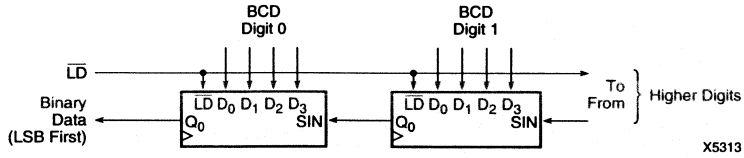


Figure 3. BCD-to-Binary Converter

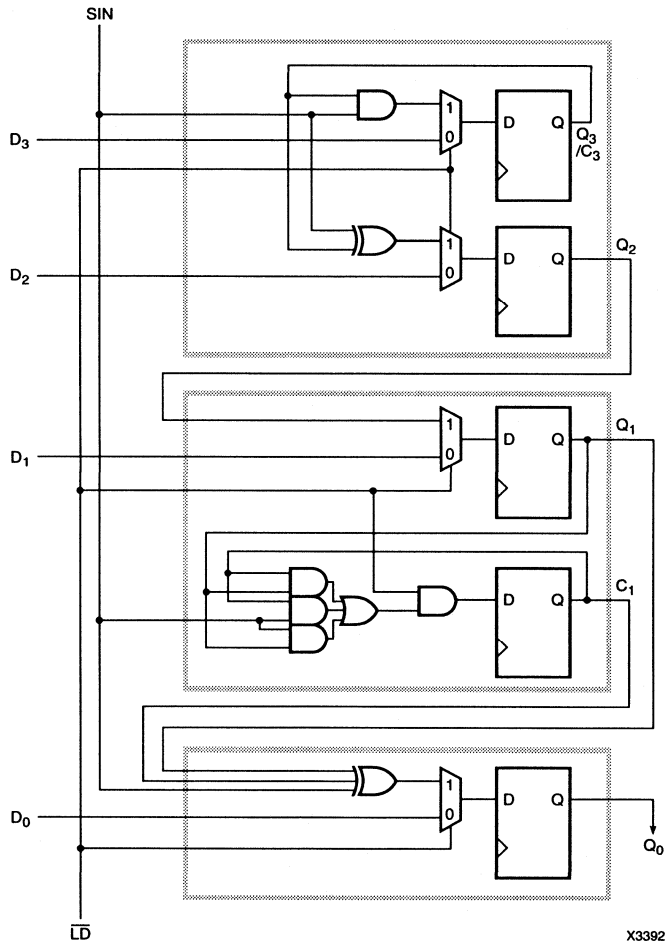


Figure 4. BCD-to-Binary Converter (Three CLBs per BCD Digit)

Frequency/Phase Comparator for Phase-Locked Loops

XAPP 028.001

Application Note By PETER ALFKE

Summary

The phase comparator described in this Application Note permits phase-locked loops to be constructed using LCA devices that only require an external voltage-controlled oscillator and integrating amplifier.

Xilinx Family

XC3000/XC3100
XC4000

Introduction

A Phase-Locked-Loop (PLL) manipulates a local voltage-controlled oscillator (VCO) so that it is in phase with a reference signal. One popular application is a programmable frequency synthesizer for radio communications. Here a crystal oscillator is divided down to a low reference frequency of 5 kHz, for example.

As shown in Figure 1, a programmable divider scales the VCO frequency down to the fixed reference frequency. The counter output is compared to the reference frequency to generate a signal that, when required, modifies the VCO frequency up or down until the comparator inputs are not only of the same frequency, but also in phase.

This frequency/phase comparator must have a wide capture range, i.e. it must generate the appropriate output, not only to pull in a small phase error, but also to correct a large frequency error. It should not generate false outputs when the input is at a multiple or fraction of the desired frequency. The well-known circuit shown in Figure 2, introduced in the early 'seventies as the Motorola MC4044, performs this function. It generates pump-up pulses when the VCO frequency is too low, pump-down pulses when its too high. The multiple feedback network assures proper operation even with large frequency errors. Figure 3 shows this circuit implemented in two CLBs plus two IOBs, directly driving the integrator (low pass filter) controlling the VCO.

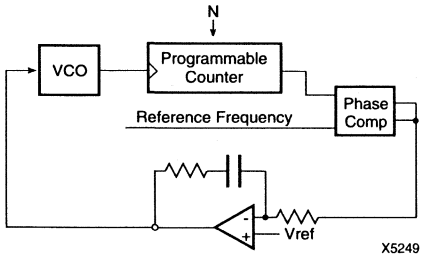


Figure 1. Typical Digital Phase-Locked Loop

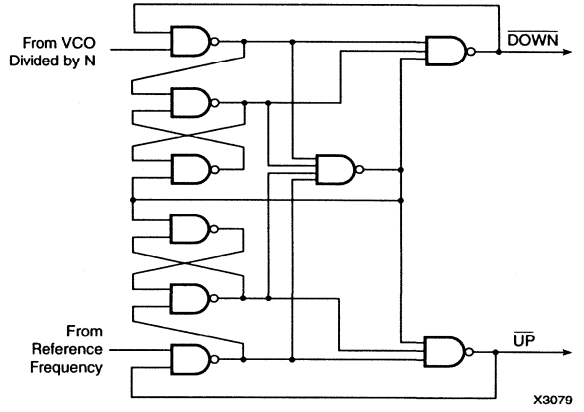


Figure 2. Digital Frequency/Phase Detector

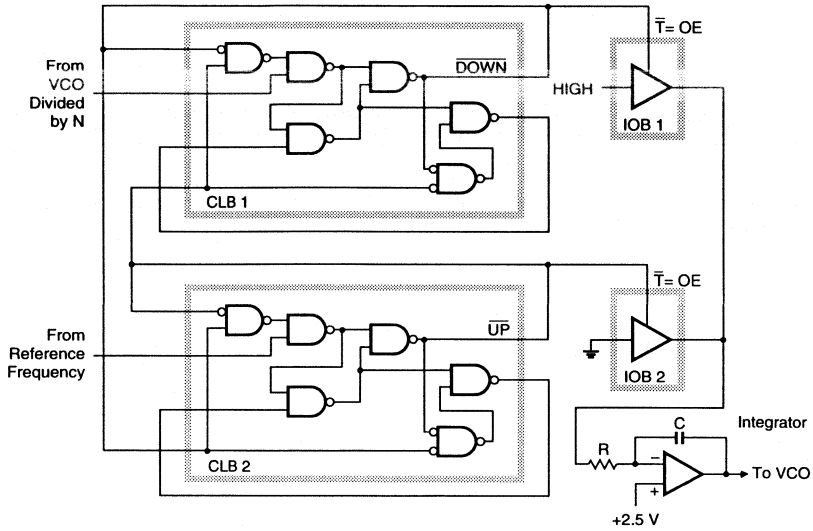


Figure 3. Frequency/Phase Detector Using Two CLBs and Two IOBs

Complex Digital Waveform Generator

XAPP 008.002

Application Note By BERNIE NEW

Summary

Complex digital waveforms are generated without the need for complex decoding. Instead, fast loadable counters are used to time individual High and Low periods.

Specifications

Minimum High/Low Time	44 ns
Maximum High/Low Time	>250 µs
Resolution	4 ns
Number of Highs and Lows	32
Number of CLBs	40

Xilinx Family

XC3000/XC3100
XC4000

Demonstrates

Fast Loadable Counters
CLB ROMs

Complex digital waveforms with unequally spaced transitions are often generated by decoding a counter that cycles with the same period as the waveform. If precise placement of edges is required, the counter must be clocked at high frequency. This increases the burden on the decoders; not only must they settle faster, but if the period of the waveform remains constant, they must become wider. These two requirements are incompatible. Decoders typically become slower as they get wider.

In LCA devices, this problem can be overcome by using high-speed counters in conjunction with data stored in ROM. The data stored in the ROM is not the waveform itself, but a run-length encoded version of it. A block diagram of the waveform generator is shown in Figure 1.

The values stored in the ROM are used to load a presettable counter that times the duration of individual High and Low segments of the complex waveform. A second counter is enabled whenever the timer is reloaded, and tracks the segment number in the waveform. This is used to address the ROM and access the length of the next segment.

The least significant bit of the second counter toggles after each cycle of the timer and thus creates the output waveform. This output is guaranteed to be glitch free, since it is generated by toggling a flip-flop.

In an LCA device, the ROM may be implemented in the CLBs. Each function generator may be used as a

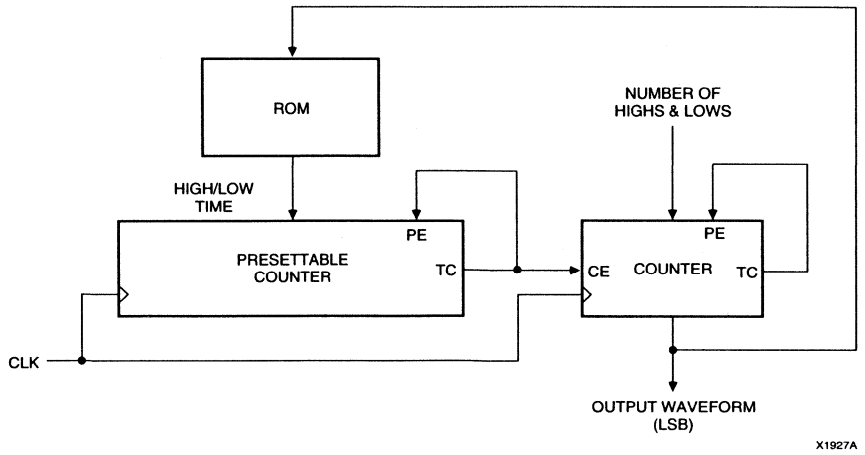


Figure 1. Precision Waveform Generator

16 x 2-bit ROM or as a 32 x 1-bit ROM. In the XC3000 series, ROM data may be entered at the schematic level using 16:1 or 32:1 multiplexers to represent ROM bits. The ROM values are applied to the data inputs of these multiplexers as hard-wired ones and zeros. CLBMAPs are used to lock the multiplexers into CLBs. APR incorporates the ones and zeros into the logic function, and creates the desired ROM as 4- or 5-input function generator look-up tables.

Using a state-skipping technique, the maximum clock rate for a presettable counter in an XC3100-series LCA device is 270 MHz. This provides for defining the duration of Highs and Lows in 4-ns increments. In such a counter, the shortest delay is 11 clocks, giving a minimum High or Low time of 44 ns. While some periods longer than this are also unavailable, the availability of all periods of 30 clocks or greater (≥120 ns) is guaranteed. The 16-bit timer allows maximum High and Low times of 262 µs. Up to 32 Highs and Lows can be accommodated using 32-word ROMs, for total waveform periods of up to 8 ms.

The 16-bit timer requires 18 CLBs, and a further six are used in the segment counter. The 32 x 16-bit ROM adds 16 CLBs, for a total of 40.

ROM values may be used more than once in a waveform. To do this, the output of the second counter must be encoded to the appropriate ROM address. With this technique, any waveform length may be accommodated, provided it comprises a limited number of distinct time intervals.

Multiple waveforms may also be generated using this scheme. A single timing counter is used to create a super-set of transition times for all the waveforms. Individual state machines are then used to create the different waveforms from this timing information.

Harmonic Frequency Synthesizer and FSK Modulator

XAPP 009.000

Application Note By BERNIE NEW AND WOLFGANG HÖFLICH

Summary

Harmonic Frequency Synthesizer
Uses an accumulator technique to generate frequencies that are evenly spaced harmonics of some minimum frequency. Extensive pipelining is employed to permit high clock rates.

FSK Modulator
A modification of the Harmonic Frequency Synthesizer that automatically switches between two frequencies in accordance with an NRZ input.

Specifications

Harmonic Frequency Synthesizer	
Maximum Output Frequency	67 MHz
Minimum Output Frequency	1 Hz
Frequency Spacing	1 Hz
Clock Frequency	67 MHz
Number of Bits	26
Number of CLBs	52
FSK Modulator	
Operating Frequencies	10/11 MHz
Jitter	±8 ns
Clock Frequency	64 MHz
Number of CLBs	10

Xilinx Family

XC3000/XC3100
XC4000

Demonstrates

Pipelining

Introduction

Most frequency synthesizers derive their output by using programmable counters to divide the clock frequency. This results in a set of attainable output frequencies that are sub-harmonics of the clock, and are defined by the following equation.

$$f_{OUT} = f_{CLK} / N$$

These frequencies are unevenly spaced, and the spacing becomes especially coarse as the required frequency approaches the clock frequency, where only one half, one third, etc. are available. If more than one exact frequency is required, the clock must be a common multiple of these frequencies.

A better approach is to use an accumulator to generate the frequencies, as shown in Figure 1. This results in a set of harmonic frequencies, defined by the equation:

$$f_{OUT} = N \times f_{CLK} / 2^n$$

Here the attainable frequencies are evenly spaced. If multiple frequencies are required, the clock need only be a binary multiple of a common factor of the frequencies. This requirement is often easier to satisfy. In particular, if the clock rate is a power of two, all integer frequencies up to the clock rate can be generated.

It must be recognized, however, that these frequencies describe the average rate at which output pulses are generated. Output transitions can only be generated an

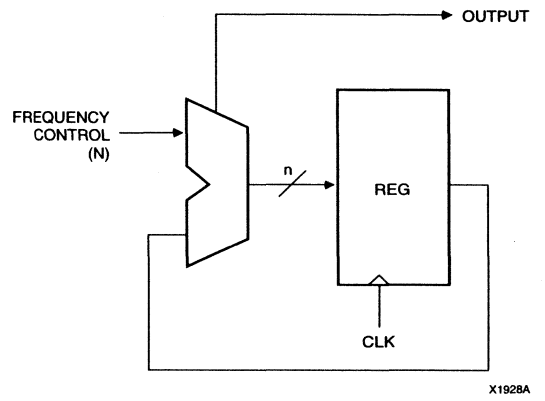


Figure 1. Accumulator-based Frequency Synthesizer

Supporting design files are available on the Xilinx Technical Bulletin Board under the name XAPP009V (Viewlogic)

integer number of clock periods apart, and this leads to jitter. As the output frequency approaches the clock rate, this jitter becomes severe.

A potential disadvantage of this scheme is the complexity of the adder and its effect on speed, when compared to the counter approach. However, this can be overcome through the use of pipelining.

Operating Description

Each Xilinx XC3000-series and XC4000-series CLB contains two flip-flops. One of these can be used to form the accumulator register, leaving the other to pipeline the carry path. A pipeline flip-flop is inserted between all the bits of the adder. The output skew this creates is not a problem as only the carry-out is of interest.

Matching the pipeline delay at the input is also not an issue if only one frequency is required, as the input never changes. If multiple frequencies are required, the input might simply be changed, but this would cause a phase discontinuity. Where this is unacceptable, a delay equalizer must be added, such that each addition into the accumulator is completed with the same input.

Conceptually, this requires a triangular array of registers, generating a 1-clock delay into the input of the second bit, a 2-clock delay into the third bit, and so on. However, this can be greatly simplified if the input only changes occasionally.

Figure 2 shows the accumulator cell with its delay equalizer. The accumulator cell is a simple full adder with its output registered and fed back to one of its inputs. A pipeline flip-flop is introduced into the carry path.

The accumulator input that controls the frequency is stored in a register. Individual bits of this holding register are enabled from flip-flops that are connected as a shift register. When the frequency is to be changed, the appropriate number is input to a holding register, and a single one introduced into the shift register. As this one propagates through the shift register, individual bits of the holding register are successively updated. This update occurs in synchronism with an addition propagating through the pipelined adder.

For an n-bit accumulator, the data must be held at the input to the holding register for n clocks after the update pulse. This is the only restriction on how fast the frequency can be changed. Also, it takes n clocks from the update pulse before the frequency change is reflected at the output. At this time, however, the change is instantaneous and phase continuity is maintained.

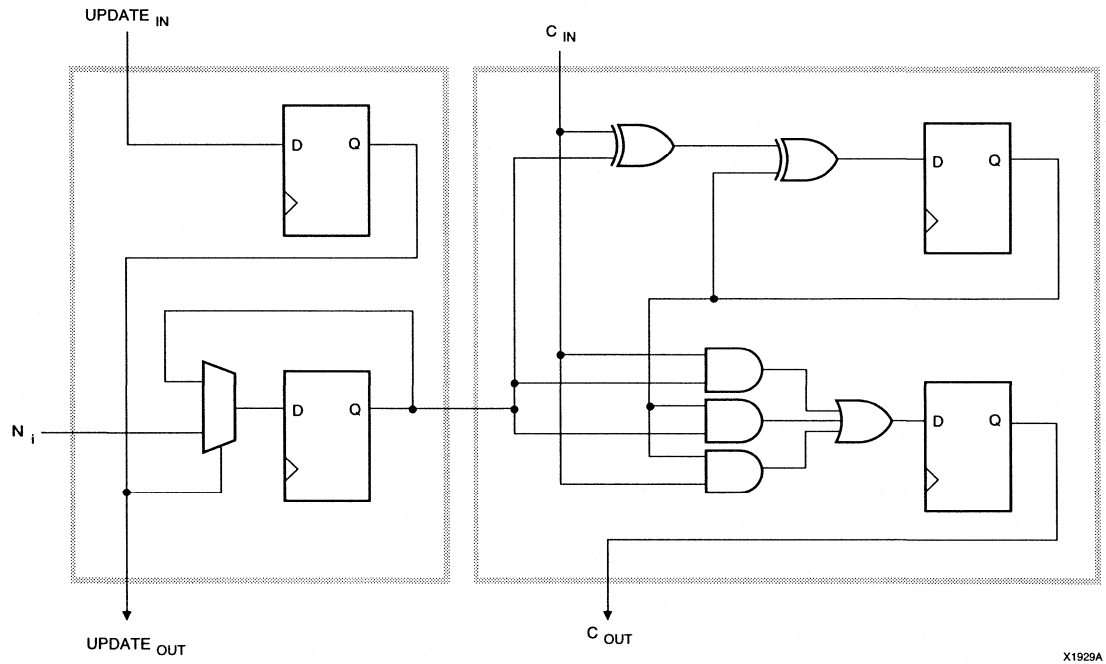


Figure 2. Bit-slice of the Frequency Synthesizer

This synthesizer design uses two CLBs per bit. Exploiting the direct interconnect between CLBs in the XC3000-series devices, either version can be operated at clock frequencies in excess of 90 MHz in a -125 part. If placement and routing do not provide for direct interconnect, the maximum speed is reduced. This is always true when the accumulator is longer than one column of CLBs in the target LCA device.

As an example, a 26-bit frequency synthesizer, clocked at 67.108864 MHz (2^{26} Hz), generates every integer-valued frequency up to this clock rate. Fifty-two CLBs are required, and the synthesizer fits into any XC3000-series LCA device.

A harmonic frequency synthesizer is useful as an FSK modulator. For FSK modulation, the synthesizer must alternate between two frequencies. This can easily be accommodated by modifying the delay equalizer, as shown in Figure 3.

Two numbers, appropriate to the two frequencies, are applied to the delay equalizer. If the frequencies must be programmable, these numbers can come from inputs or registers. Otherwise, they can be hard-wired at the inputs of the CLBs, or individual function generators can be modified to incorporate them.

NRZ data is applied to the shift register. As this propagates through the shift register, multiplexers at the input to each bit of the holding register detect changes in the data. When a change is detected, the bit is reloaded from the appropriate number. Again, changes ripple through the holding register in synchronism with the additions. The NRZ data may change every clock, if required.

A typical FSK modulator, as shown in Figure 4, might be required to switch between 10 and 11 MHz. To give a square output, a flip-flop is used to divide the synthesizer output by two. This flip-flop may be the carry pipeline of the final adder, modified to toggle with the carry rather than storing it.

The toggle flip-flop must be enabled at frequencies that are twice the output frequencies. The largest common factor of 20 and 22 MHz is 2 MHz, and the clock frequency must be a binary multiple of this. Higher binary multipliers will result in lower jitter. In this case, 64 MHz is chosen. This is 2^5 times 2 MHz, and a 5-bit accumulator must be used. Twenty and 22 MHz are 10 and 11 times 2 MHz, respectively, and these are the numbers that must be accumulated to generate the frequencies (0A Hex and 0B Hex). This FSK modulator may be implemented in only 10 CLBs.

If an analog output is required, either version of the synthesizer may be used to control a counter, as shown in Figure 5. The output of this counter is used to access a look-up table, which provides data to a DAC.

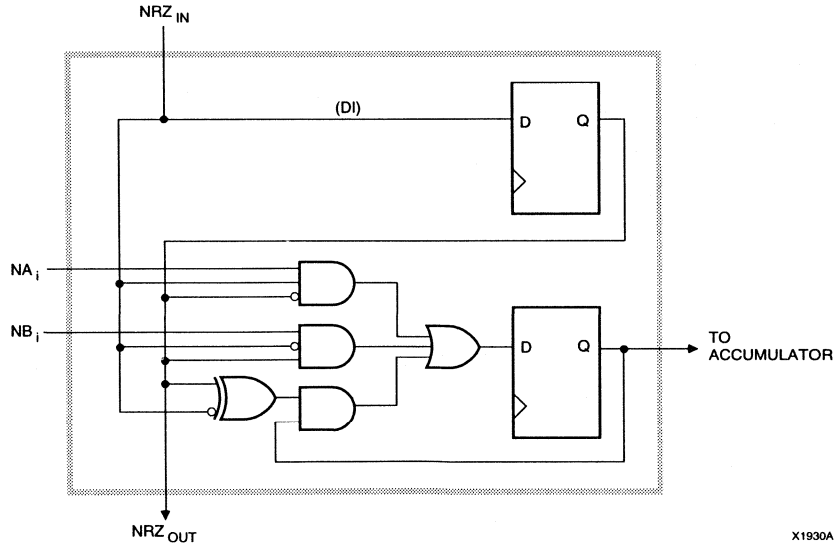


Figure 3. Delay Equalizer for an FSK Modulator

In the XC3000-series, the CLB function generators may be used as ROMs to implement the look-up table internally. The CLBs actually contain RAMs that are written during configuration. As a result, multiple wave-shapes can be supported by re-configuring the LCA device. The XC4000-series provides user-accessible RAM in the CLB. Wave-shapes, therefore, can be changed on the fly.

An external look-up table may also be used. In particular, a video RAMDAC can be loaded with the wave-shape. This is sequentially addressed at appropriate intervals to generate the waveform with the desired frequency.

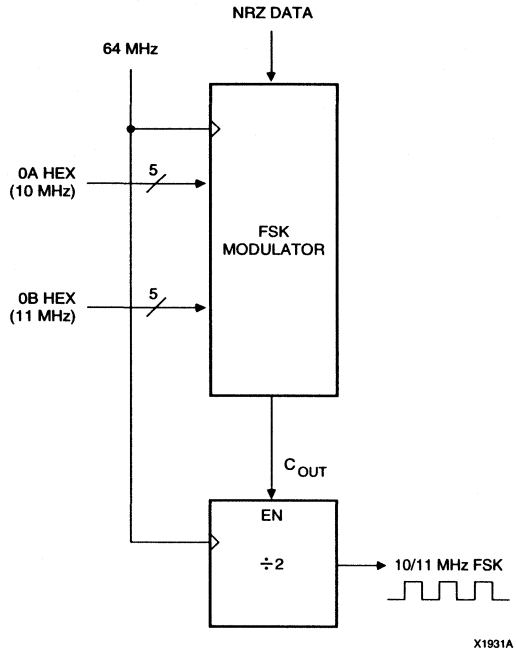


Figure 4. 10/11-MHz FSK Modulator

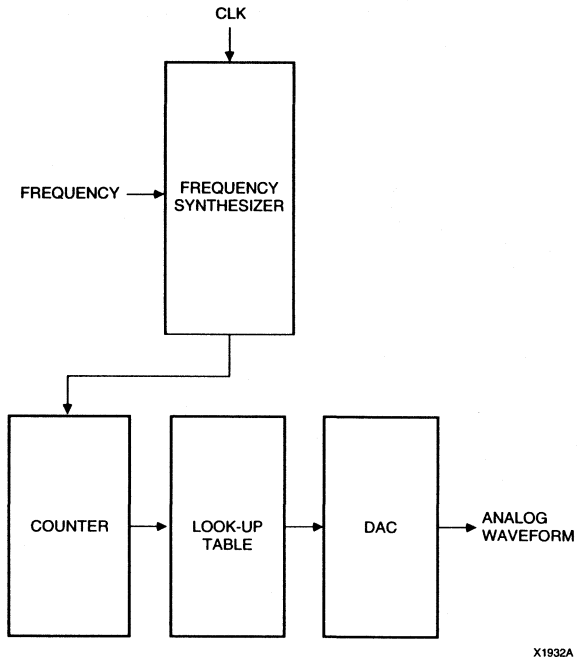


Figure 5. Analog Waveform Generator

Implementing State Machines in LCA Devices

XAPP 027.001

Application Note BY PETER ALFKE AND BERNIE NEW

Summary

This Application Note discusses various approaches that are available for implementing state machines in LCA devices. In particular, the one-hot-encoding scheme for medium-sized state machines is discussed.

LCA Family

XC3000/XC3100/XC3100A

Demonstrates

State Machine Design
One-hot Encoding

Introduction

State-machine methodology defines the contents of every flip-flop in a design under every circumstance that might arise. It also defines all the possible transitions that can cause the design to go from one of these states to another. In its simplest form, this is just a rigorous way of designing synchronous logic, like 4-bit counters. For more complex designs, the state-machine approach gives the designer a tool to analyze all possible operating conditions, and so avoid overlooked hang-up states or undesired transitions. LCA devices with their abundance of flip-flops lend themselves well to state-machine designs.

Using the 5-input function generator of the XC3000 family devices as a 32-bit ROM, a state machine with up to 32 states with no conditional jumps uses only five CLBs. Five registered CLB outputs drive the five function generator inputs of five CLBs in parallel. This implements a fully programmable sequencer such as a synchronous counter.

For a smaller number of states, some inputs can be used as conditional jump inputs. Encoding these condition codes, however, may require an additional level of logic which reduces the maximum clock rate.

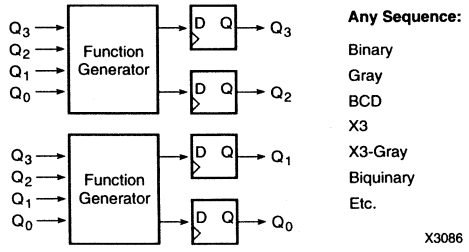


Figure 1. Synchronous 4-Bit Counter in 2 CLBs

Synchronous Counters

Using only two CLBs, it is possible to construct fully synchronous 4-bit counters with arbitrary count sequences, Figure 1. The CLB Clock Enable inputs even provide count-enable control. The count length, count direction, and even the code sequence is determined by the configuration. The number of possible count sequences is factorial 15, i.e., more than 10^{12}. All four outputs are available, and while the counter cannot be preset to an arbitrary value, it can be cleared by an asynchronous input.

Table 1 shows four common count sequences. Of particular interest is the Gray code, which offers glitch-less decoding, since only one bit changes on any transition. A Gray-code counter can also be reliably read asynchronously. In contrast, if a binary counter is read during its transition between 7 and 8, for example, any code might be detected.

Decimal	Binary	Gray	X3 Binary	X3 Gray
0	0000	0000	0011	0010
1	0001	0001	0100	0110
2	0010	0011	0101	0111
3	0011	0010	0110	0101
4	0100	0110	0111	0100
5	0101	0111	1000	1100
6	0110	0101	1001	1101
7	0111	0100	1010	1111
8	1000	1100	1011	1110
9	1001	1101	1100	1010
10	1010	1111		
11	1011	1110		
12	1100	1010		
13	1101	1011		
14	1110	1001		
15	1111	1000		

Table 1. Four Common Binary Count Sequences

Four-bit counters constructed as described above can easily be concatenated into longer, four-bits-at-a-time ripple-carry counters. For each 4-bit digit, a third CLB is used to detect an arbitrary terminal count value, and AND this with the incoming Count Enable to provide the Count Enable to the next digit.

Waveform Generator

Arbitrary binary waveforms of any length up to 32 clock periods can be generated using only three XC3000 series CLBs, Figure 2. The waveform generation is fully synchronous, and may be paused at any time, using the CLB Clock Enable. It may also be restarted, using the asynchronous clear.

Five flip-flops, Q_{0-4}, form a linear feedback shift-register counter. The 5-input combinatorial function generator, F_0, determines both the modulus and the count sequence; there are no illegal or hang-up states. The function generator, F_1, operates as a ROM, and can be programmed to provide any conceivable decode of the counter. Flip-flop, Q_5, synchronizes and de-glitches the decoder output.

The following examples demonstrate the arbitrary nature of the waveforms that can be generated.

Example 1. + 28 counter with its output High at times T2, T3, T10, T22 through T27

Example 2. +19 counter with its output Low at times T9, T12, T15, T18.

Simple State Machines

The simple state machine shown in Figure 3 uses only 10 CLBs, and has up to 16 states. Each of eight outputs decode/encode any combination of states. The state machine is based on a 5-CLB next-state look-up table.

Each state corresponds to two look-up table locations that store two arbitrarily defined next states. From any state, the C input controls a two-way branching by selecting which of the two possible next states is asserted. For hold loops, one of the next states should be the current state; and to avoid branching, both destination states should be made equal.

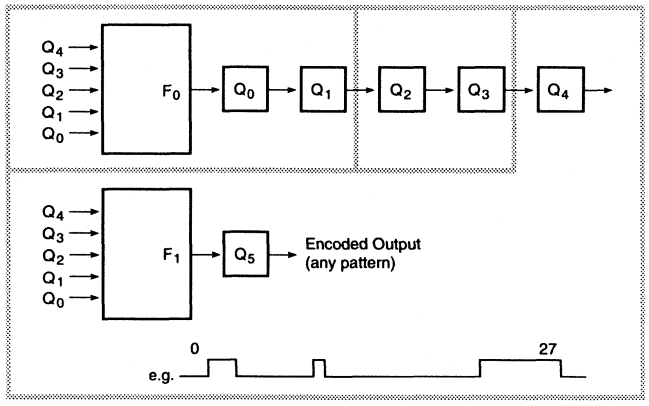


Figure 2. Synchronous 5-Bit Waveform Generator in 3 CLBs

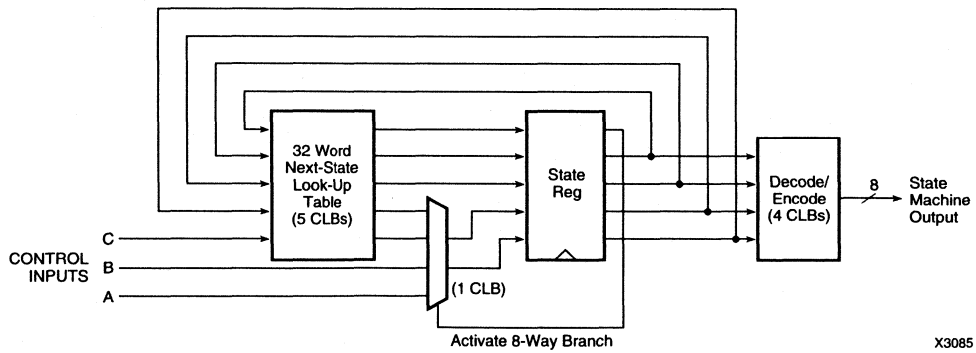


Figure 3. Simple Sate Machine

The state machine can also perform 8-way branches from any state so programmed. The branch destinations must all fall in two quadrants (0..3, 4...7, 8...11 or 12...15). The choice of the two quadrants is arbitrarily programmed into the look-up table; C selects between the two quadrants, and A and B select the state within the quadrant.

Activation of the 8-way branch mechanism is controlled by a fifth state bit that is set during the transition into the state. This bit controls a multiplexer that replaces the two LSB of the destination state with the control inputs A and B. Note that as the fifth bit is independent of A and B, it must be set, or not, on a per quadrant basis during an 8-way branch.

Examples:

- From state 3, if C = High, go to 5, else go to 8
- From state 7, if C = High, go to 3, else stay in 7
- From state 9, unconditionally go to 2
- From state 6, execute the truth table below

Truth Table

A	B	C = Low	C = High
0	0	12	0
1	0	13	1
0	1	14	2
1	1	15	3

One-Hot Encoded State Machines

The state machines described have encoded state bits. For an N-state state machine, fewer than N flip-flops are used (but $\geq \log_2 N$), and a unique combination of these flip-flops is set in each state; each flip-flop is set in several states. While this minimizes the number of flip-flops, it increases the complexity of the logic controlling each flip-flop.

In LCA devices, flip-flops are plentiful, and there is no need to conserve them. Consequently, for medium-sized state machines, it is better to use a One-Hot encoding scheme (OHE). OHE increases the number of flip-flops required, but reduces the logic complexity associated with each of them, thereby boosting performance.

In an OHE state machine, one flip-flop is assigned to each state. It is set during that state, and only during that state. The state machine is implemented as a shift-register-like structure, where a single One is passed from flip-flop to flip-flop, sometimes holding in the same flip-flop, skipping bits of the shift register or moving to a parallel shift register, Figure 4a and b.

The control logic associated with each state bit involves ORing the transitions into the state, including any hold loop. Each of these transitions will involve a previous state, which, by design, is represented by a single bit. This bit may, or may not, be ANDed with some decode of the control bits inputs.

It is the localization of the control logic that leads to the performance increase. For each state bit, the control logic only involves the limited number of state bits from which there are transitions and the conditions that control those transitions. This permits shallow logic structures between flip-flops, often only requiring the function generator associated with the state-bit flip-flop. In addition, no state decoding is necessary, and state encoding can only require the ORing of state bits.

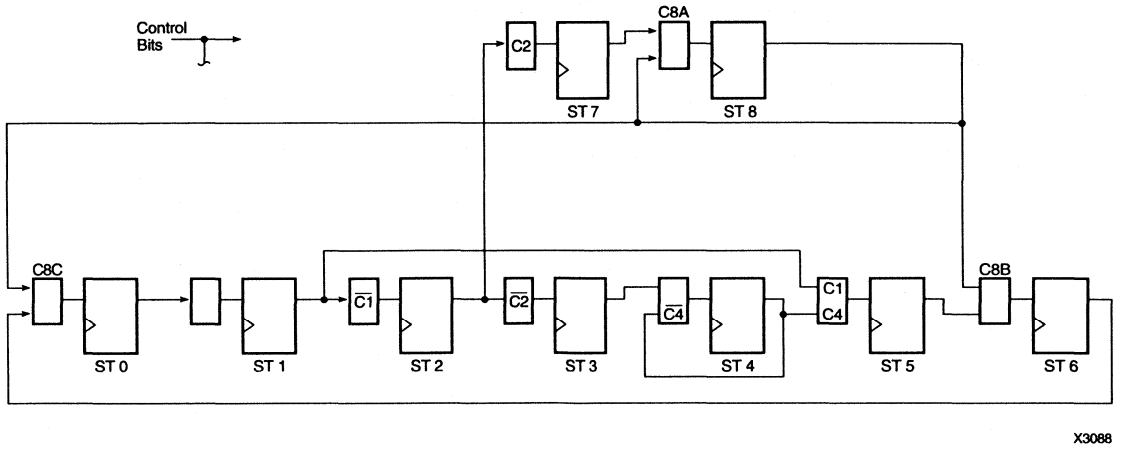


Figure 4a. Prototype OHE State Machine

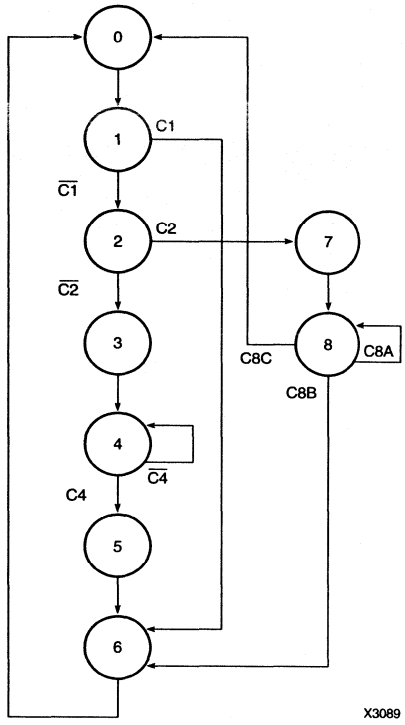


Figure 4b. State Diagram for Prototype OHE State Machine

Complex State Machines

Small- and medium-sized state machines can easily be implemented within an LCA device, as shown above. For large, complex state machines, however, it is better to use the LCA device to implement a simple microsequencer, and store the control program externally, Figure 5.

For fastest operation, a high-speed SRAM should be used for the control program. This may be loaded from a microprocessor, or shadowed by an EPROM. For slower operation requiring non-volatility, an EPROM can be used directly. When an EPROM is used, the number of components can be reduced by storing both the LCA configuration data and the state-machine control program in the same device.

If an XC3020 is configured in the Master Parallel mode and it reads its configuration data out of a 256K (32K x 8) EPROM, it only requires 6% of the addresses, from the top location 7FFF (32K) through 77FF (about 30K). The remaining 94% of the EPROM can be used as a next-state look-up table with a capacity of 240 states.

Eight state bits are read out of the EPROM and registered in the LCA device which can perform any required decoding

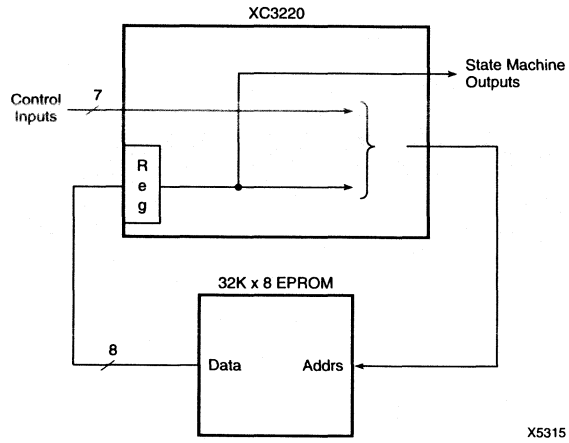


Figure 5. Rudimentary Complex State Machine

or encoding of the state-machine outputs. The registered state bits also form part of the new EPROM address, defining a block of 128 possible next states. The 7-bit condition code completes the EPROM address and selects which of 128 next states is actually asserted.

Each transition is, in effect, a 128-way branch. However, the branching complexity will normally be reduced by assigning identical values to many of the 128 possible next states.

Since the top 16 address locations are used for configuration data, the state codes, which form the 8 MSBs of the EPROM address, are limited to 240 different values, 0...239. The control inputs provide the seven LSBs of the EPROM address. If the control inputs are asynchronous, they must be registered for reliable operation.

This rudimentary state machine can thus have 240 different states, and can jump from any state to any one of 128 arbitrarily defined next states, according to a 7-bit condition code. In its simplest form, this basic design consumes no CLB resources in the LCA, just IOB flip-flops for the state register. Even so, it permits a number of states and a multiway branch complexity far in excess of any normal need.

The user has all the logic resources of the LCA available to add features like the following.

- State decoding/encoding
- Stack registers
- Loop counters
- More sophisticated branch logic, etc.

This design is straightforward, inexpensive, compact and extremely flexible. Its speed is limited primarily by the control store access time; faster access times can be obtained using SRAMs in place of EPROMs.

Light-Driven Counter Controller

XAPP 012.001

Application Note By PETER ALFKE AND BERNIE NEW

Summary

A simple state machine is used to adapt the output of two photo-cells to control an up/down counter. The state machine provides hysteresis for counting parts correctly, regardless of changes in direction.

Specifications		Xilinx Family
Maximum Clock Frequency	~150 MHz	XC3000/XC3100
Number of CLBs	2	*Demonstrates*
		State-Machine Design

Introduction

A common technique for counting objects is to pass them through a light beam. Problems can arise, however, if a part dithers on the edge of the light beam and is counted more than once, or if the direction of motion changes and a part is recounted rather than uncounted.

These problems may be avoided by using two sensors, as shown in Figure 1. To be counted, an object must first obscure one sensor, then obscure the other, clear the first and finally clear the second. This solves the dither problem as an object must pass entirely through the beam before it can be counted. Sensor signals resulting from the object dithering while entering or leaving the beam will be ignored by the counter.

The direction of motion determines the order in which the sensors are first obscured and then cleared. A state machine recognizes the order and controls an up/down counter to correctly account for parts that pass back and forth through the beam. The hysteresis in the state machine even accommodates directional changes while a part is in the beam.

For the scheme to operate correctly, the object must be large enough to obscure both sensors. The sensors are used to control a synchronous state machine, and the object must move slowly enough that it does not obscure or clear both sensors within one clock period.

The bidirectionality of this scheme also makes it suitable for position sensing. The objects discussed above are replaced by a comb attached to some moving part. The part position is determined by counting the teeth on this comb as they pass through the light beam.

Operating Description

The state diagram of the counter controller is shown in Figure 2. Inputs A and B are High when the sensors are obscured. While no objects are present, the state machine holds in the Wait state. As an object moves into the beam, state variables S1 and S2 simply follow the inputs with a one clock delay. When the object exits the

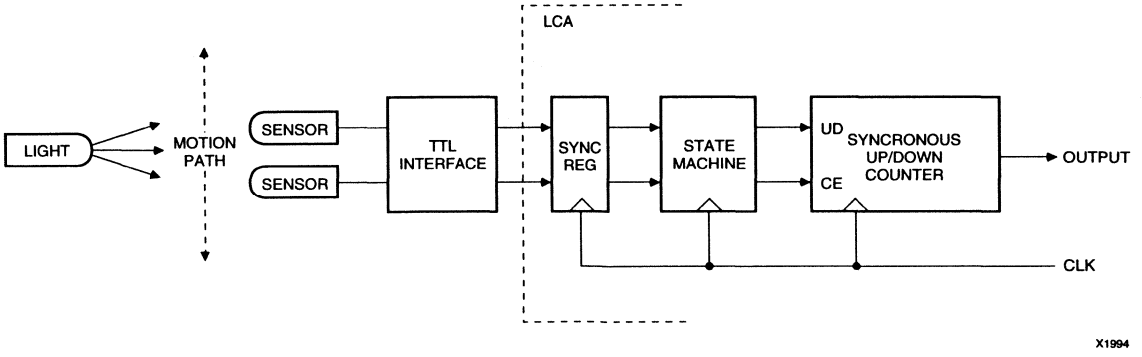
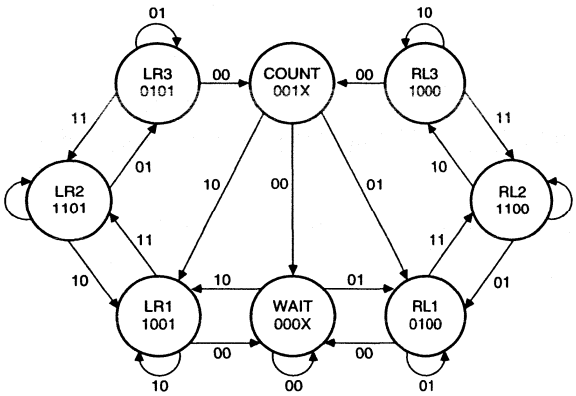


Figure 1. LCA Light-driven Counter



$S_1 = A$

$S_2 = B$

$S_3 = \overline{S}_1 \bullet S_2 \bullet S_4 \bullet \overline{A} \bullet \overline{B}$ (COUNT ENABLE)
$+ S_1 \bullet \overline{S}_2 \bullet \overline{S}_4 \bullet \overline{A} \bullet \overline{B}$

$S_4 = \overline{S}_1 \bullet \overline{S}_2 \bullet A$ (UP/DOWN)
$+ \overline{S}_1 \bullet \overline{S}_2 \bullet S_4$

Input State Variables	AB S_1 S_2 S_3 S_4

X2653

Figure 2. Light-driven Counter State Diagram

beam, the Count state is entered (S3 High) and the counter is enabled. One clock later, the state machine automatically moves out of the Count state and into the Wait state. If a new object is sensed from either direction during the Count state, the Wait state is omitted and the appropriate sequence commenced.

The identity of first sensor to be obscured is stored as the S4 variable. This is used to determine which sensor must be cleared last to ensure that the object has cleared the beam without reversing its direction. S4 also selects up or down operation of the counter. The up/down control is set up at least four clocks before the counter is enabled.

The state machine can be implemented in three CLBs, as shown in Figure 3. The asynchronous TTL-level signals are brought into the LCA device and registered in the IOBs. This synchronizes them to the state-machine clock and eliminates any metastability problems. S1 and S2 share a CLB. S3 is a function of five variables and requires a whole CLB. S4 occupies the third.

If required, the state machine implementation can be reduced to two CLBs. Using the DIN input, S1 can be combined with S3. S2 can then share the second CLB with S4.

Any synchronous up/down counter design may be used in conjunction with this state machine. The maximum count rate required is one fourth the clock rate.

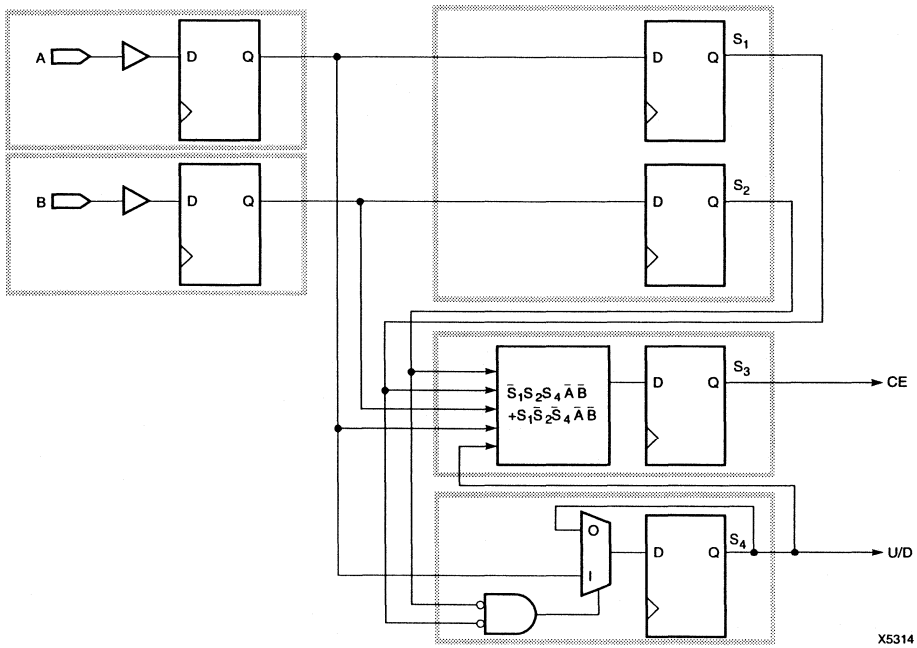


Figure 3. Light-driven Counter State Machine

Designing Complex 2-Dimensional Convolution Filters

XAPP 037.000

Application Note By DAVID RIDGEWAY

Summary

This Application Note shows how to design complex 2-dimensional filters for digital image processing systems. The XC7200/XC7300 dedicated carry logic is used to perform the complex arithmetic functions.

Xilinx Family

XC7200/XC7300

Demonstrates

High-Performance Arithmetic

Introduction

A digital-image-processing system can acquire an image of an object, process or modify the image data, and use the result in the performance of a task. In such imaging systems, edge detection is fundamental to obtaining such information as contrast, shape, location, and dimension. However, conditions can occur that make the true image edges difficult to detect.

To improve the image quality so that edges can be more accurately identified, image processing systems use digital filtering. This process creates a new image where the data is altered to enhance features of interest.

The performance of digital-image-processing filters is usually limited by software algorithms and system throughput. Faster speeds can be achieved with modified algorithms and dedicated hardware. Using the high-speed arithmetic logic functions embedded into the XC7200 architecture, image-processing systems can perform computationally intensive tasks, such as edge detection and enhancement, without burdening the processor. This feature significantly improves the overall system performance by maximizing the computational throughput.

Two-Dimensional Convolution

Two-dimensional convolution is a common digital image filtering technique. A new value is calculated for each pixel in the image, based on the value of the corresponding pixel in the old image and those that surrounded it.

In industrial applications, a popular filter operator is the Laplacian edge-enhancement operator, as illustrated in Figure 1. A 3 x 3 coefficient matrix is overlaid on the image, and the nine pixels it covers are each multiplied by the corresponding coefficient. The sum of the nine products is the value in the new image of the pixel that corresponds to the center of the matrix in the original image. For example, if the matrix is centered over the corner [1] in the figure, the result is the [5] in the output image.

Image Data		Laplacian Filter		Detected Edge
0 0 1 1 1				0 -3 3 0 0
0 0 1 1 1		-1 -1 -1		0 -3 3 0 0
0 0 [1] 1 1	X	-1 8 -1	=	0 -2 [5] 3 3
0 0 0 0 0		-1 -1 -1		0 -1 -2 -3 -3
0 0 0 0 0				0 0 0 0 0

$$[5] = P_{1,1} X f_{1,1} + P_{2,1} X f_{2,1} + \ldots P_{3,3} X f_{3,3}$$

Figure 1. Laplacian Edge-Enhancement Operator

The Laplacian operator is particularly appealing since all the coefficients are binary powers. Consequently, the multiplications can be replaced by shifts, which greatly simplifies the operation and increases throughput.

The process is repeated with the coefficient matrix centered over each pixel in turn, until new values have been obtained for each pixel in the image. Effectively, the operator differentiates the image. There is an increase in magnitude and a sign change in the vicinity of an edge, and in areas where there is no edge, the output is zero. After convolution, the data is scaled by a factor of 9, negative values are rectified, and background information is discarded.

XC7200 Dedicated Carry Logic

XC7200-series Macrocells contain dedicated, hard-wired carry logic that accelerates and condenses arithmetic functions such as adders and accumulators. Macrocells are organized into Function Blocks, each containing nine Macrocells. The dedicated logic propagates carries between adjacent Macrocells and adjacent Function Blocks. This feature makes it possible to develop fast, wide arithmetic functions. Adders can achieve ripple-carry delays as low as 1.0 ns per bit.

A detailed schematic diagram of the Macrocell dedicated carry logic is shown in Figure 2. The arithmetic logic unit (ALU) is a 2-bit function generator that can be programmed to generate any Boolean function of the D_1 and

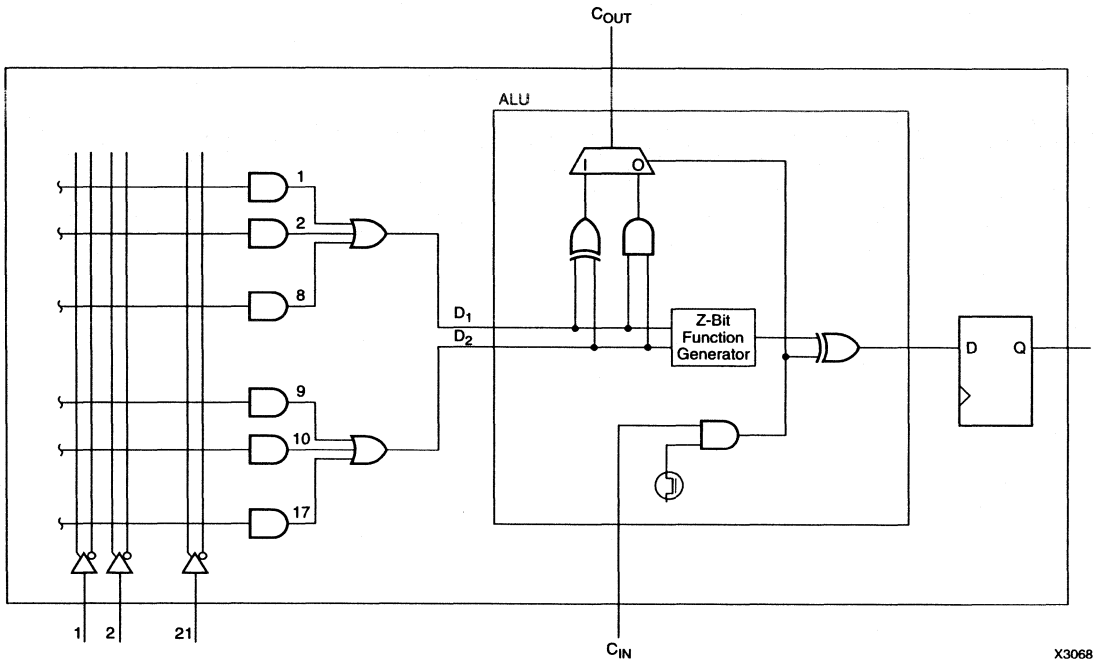


Figure 2. Macrocell With Detailed Carry Logic

D_2 inputs. Combined with the carry input (C_{IN}) signal from the lower Macrocell, the ALU can generate either the arithmetic sum or difference of two operands, and the carry output (C_{OUT}) to the next higher Macrocell.

Operating Description

Image Convolution

Complete images can be processed at high speeds using a pipelined algorithm that exploits the architectural features of the XC7200/XC7300 EPLD family. The complete Laplacian edge-enhancement filter design is shown in Figure 3.

Image data is input line-by-line. Two shift registers, each one line long, delay the incoming data such that corresponding pixels from each three consecutive lines are available simultaneously. The EPLD stores three consecutive pixels from each line in three shift registers, thus making available a 3 x 3 array of pixels.

These nine pixels are selected in turn as the input to the accumulator. A second multiplexer at the input to the accumulator performs the trivial multiplications. Pixels to be multiplied by eight are shifted three places towards the most significant bit (MSB); pixels to be multiplied by -1 are inverted and a carry forced into the accumulator completes the subtraction.

A 1-bit slice of the Multiplier-Accumulator is shown in Figure 4.

Filter Performance

As with any EPLD design, performance can be estimated with complete accuracy prior to implementation. The data throughput rate is limited by the propagation delay of the carry chain from the least significant bit (LSB) input to the MSB output for the multiply-accumulate function. For the Laplacian filter design implemented in an XC7272-25 with 8-bit pixels, the maximum propagation delay is approximately the following.

$$t_{PD} = 25 + 15 \text{ ns}$$

This gives an accumulation rate of 25 MHz, and a convolution output rate of 2.5 MHz. For a 512 x 512 image (262,144 pixels), the convolution time for one frame is 104 ms, which is equivalent to 9.6 fps. Higher speed image convolutions can be achieved by using multiple pipelined accumulators and summing the output data. In a fully pipelined design, the same 512 x 512 image can have a convolution time of 10 ms and a frame rate of 100 Hz.

References

1. Louis J. Galbiati, Jr., Machine Vision and Digital Image Processing Fundamentals, Englewood Cliffs: Prentice Hall, 1990.

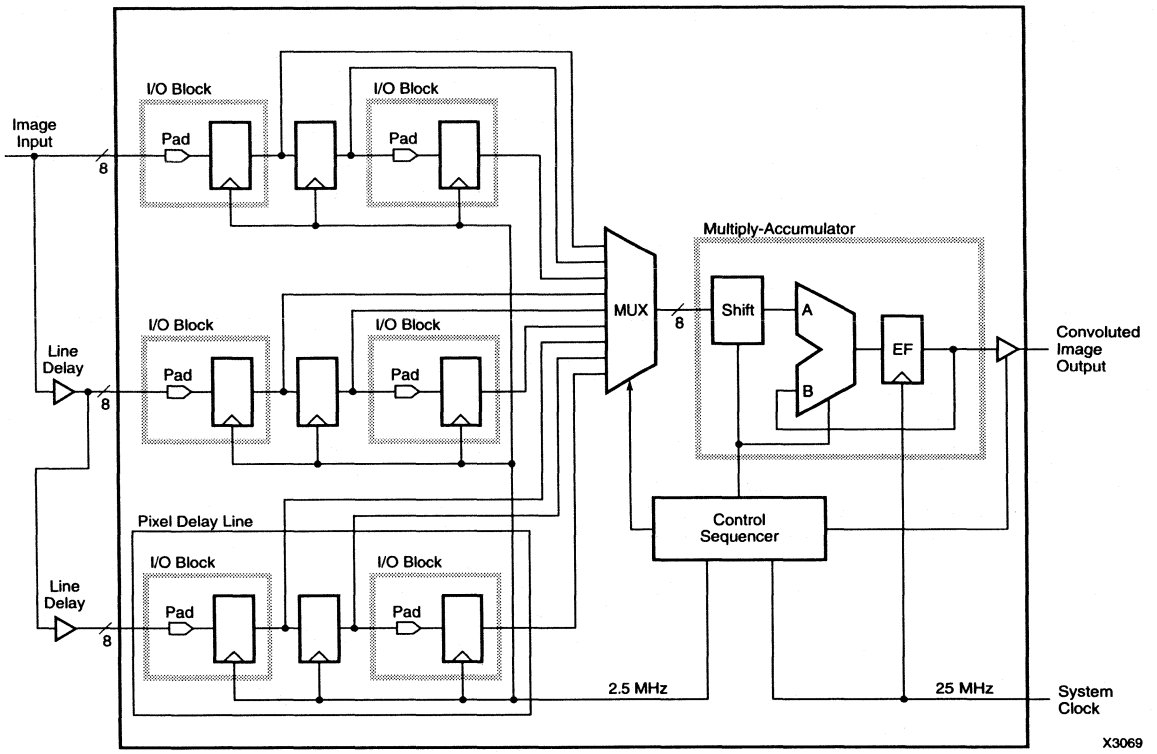


Figure 3. Laplacian Edge-Enhancement Filter

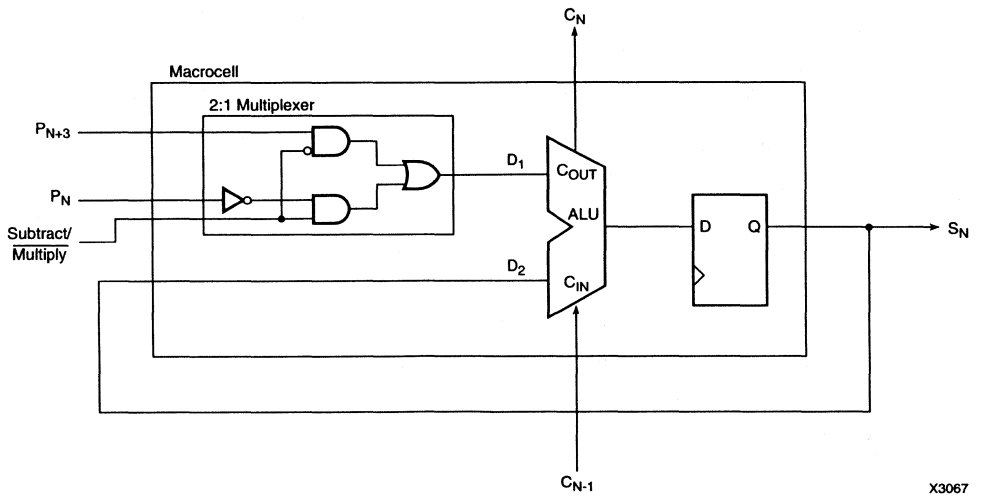


Figure 4. Bit-Slice of the Multiply-Accumulator

Four-Port DRAM Controller Operates at 60 MHz

XAPP 036.001

Application Note By JEFFREY GOLDBERG

Summary

This Application Note describes a high-performance DRAM controller implemented in a single Xilinx EPLD.

Xilinx Family

XC7200/XC7300

Demonstrates

High-speed State Machines

Introduction

Multi-port memory arrays are used in many applications, such as telecommunications, graphics and VME cards. Although these applications serve many different purposes, they share a common need: they must quickly and efficiently access a shared memory space through several different ports. The control logic must perform a complex arbitration function, yet must run at a high clock speed.

The XC7236A architecture is well suited for implementing the fast, complex state machines found in multi-port arbitration schemes. The XC7236A-16 can implement a quad-ported DRAM memory controller capable of arbitrating among four access requests in one 60-MHz clock cycle. This DRAM controller is capable of supporting 70-Mbyte/s burst transfers over a 32-bit bus, Figure 1.

The design uses 94% of the available Macrocells, yet runs at the maximum specified speed of the device. Familiar third-party tools reduce both the design effort and time, and XEPLD translator quickly compiles even the most complex designs.

Theory of Operation

The arbiter implements a round-robin algorithm, where the priorities for the four ports are arranged in circular order; the most recently served port is automatically assigned lowest priority. Each port can also lock the arbiter to retain ownership between back-to-back accesses. Such locking is necessary for semaphore reading and writing in multiprocessor systems.

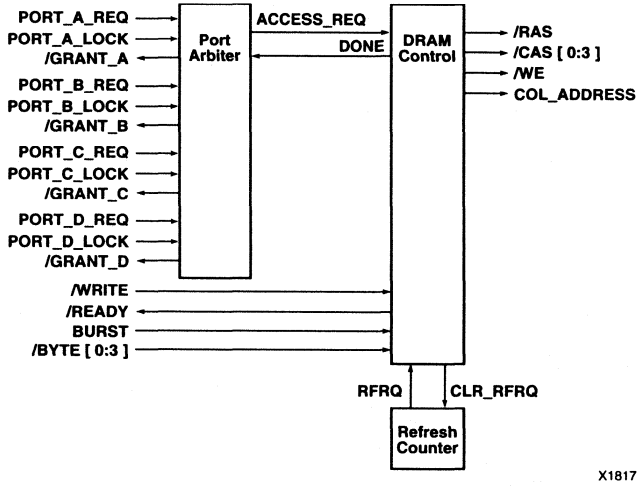


Figure 1. Quad-Port Memory Controller

The arbiter evaluates incoming access requests while it is in any of four idle states. The specific idle state depends on the last request, and determines the priority of the incoming requests. If the arbiter is not locked, it grants access to the highest priority request that is pending, and issues a memory-access request to the on-chip DRAM controller. During its transition to one of four port-access-active states, the arbiter asserts the grant signal to the appropriate port. The grant signals are used to enable the port control, address and data busses. The arbiter remains in its active state until the DRAM controller signals that it has completed the single or burst access.

The arbiter then goes to the idle state corresponding to the port that was just serviced, thus placing that port at the lowest priority level. If another access request is pending, the arbiter will issue another memory-access request to the DRAM controller. The data access will occur as soon as the DRAM controller has precharged the memory. The interaction between arbiter and DRAM controller is shown in Figure 2.

The DRAM controller also arbitrates between memory requests from the ports and refresh requests from the on-chip refresh counter, as can be seen in Figure 3. The address-multiplexer control line and the DRAM strobes are sequenced by the controller's state machine. They are enabled by the byte select and write enable outputs of the port.

The controller informs the port when there is valid data on the bus by asserting the READY output. If burst access is enabled, fast 3-clock memory accesses are performed until the port drops the burst request line. The controller then begins to precharge the memory, and asserts DONE to inform the port arbiter that the final memory access is completed.

Device Utilization

When implemented in PLDs, multi-port-arbiter state machines tend to be product-term intensive. The XC7236A is particularly well suited for such applications since each Macrocell can handle up to 17 product terms.

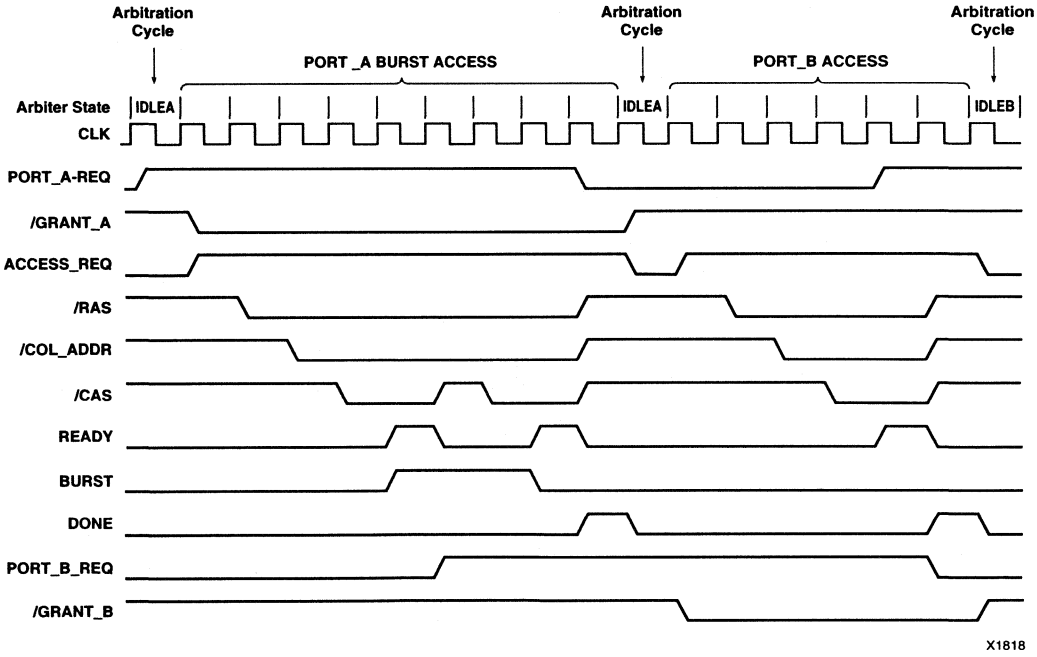


Figure 2. Quad-Port DRAM Controller Timing Diagram

Of the eight Macrocells required to implement the port arbiter, one Macrocell uses ten product terms, one uses nine terms, one uses eight terms; the remaining five Macrocells use seven product terms each. In total, 148 product terms, and 34 of the 36 Macrocells are used. The Macrocell XOR gates in the XC7236 significantly reduce the number of product terms used in the 10-bit refresh counter. In total, the DRAM controller occupies 94% of the XC7236A.

Design Methodology

The design lends itself very well to a modular behavioral description of its state machine. The ABEL 4 compiler was used to generate three Boolean equation files from high-level descriptions of the refresh counter, and the arbiter and DRAM controller state machines. A main PLUSASM file was then derived from the three equation files.

In this design, the main file only defines the external signals to and from the XC7236A. With this design approach, individual state machines and counters can be developed in a modular fashion, using the design tools most appropriate to each module. XEPLD software compiles the files in about five minutes, and generates a single file that can be downloaded into the device programmer.

Detailed design files are included with the XEPLD software, and are available from Xilinx Applications. They will soon be available from the Xilinx Technical Bulletin Board.

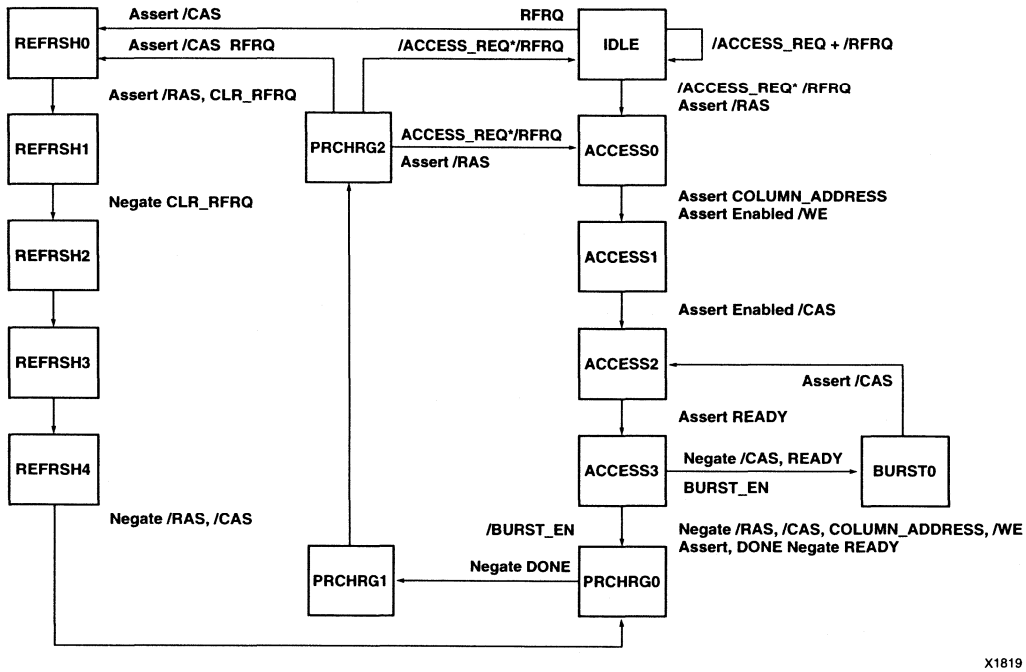


Figure 3. DRAM Controller State Diagram

Bus-Structured Serial Input/Output Device

XAPP 010.001

Application Note By BERNIE NEW

Summary

Simple shift registers are used to illustrate how 3-state busses may be used within an LCA device. Dedicated wide decoders are used to decode an I/O address range and enable the internal registers.

Specifications

Bus Width	16 Bits
Maximum Bus Speed	40 MHz
Number of Serial Channels	12
Maximum Serial Speed	60 MHz
Number of CLBs	96

Xilinx Family

XC4000

Demonstrates

3-state Buffers
Wide Decoders

Introduction

The combination of long data lines and 3-state buffers, found in Xilinx devices, is ideal for bus-structured applications. In this simple example, multiple shift registers are implemented to provide a serial input and output facility. This is purely illustrative, and the shift registers may easily be replaced with more complex functions.

In an XC4000-series LCA device, there are two horizontal Longlines equipped with 3-state buffers (T-BUFs) between each row of CLBs. In an XC4005 that has 14 rows of CLBs, there are 28 such lines. However, each of these may be split into two independent halves. This provides for construction of up-to-56-bit busses, although the number of potential bus connections is reduced.

For the purposes of this example, shown in Figure 1, a 16-bit bus is created. The flip-flops in the CLBs are used to implement the shift registers, with two bits per CLB (eight CLBs per shift register), as shown in Figure 2. The function generators preceding the flip-flops are used to

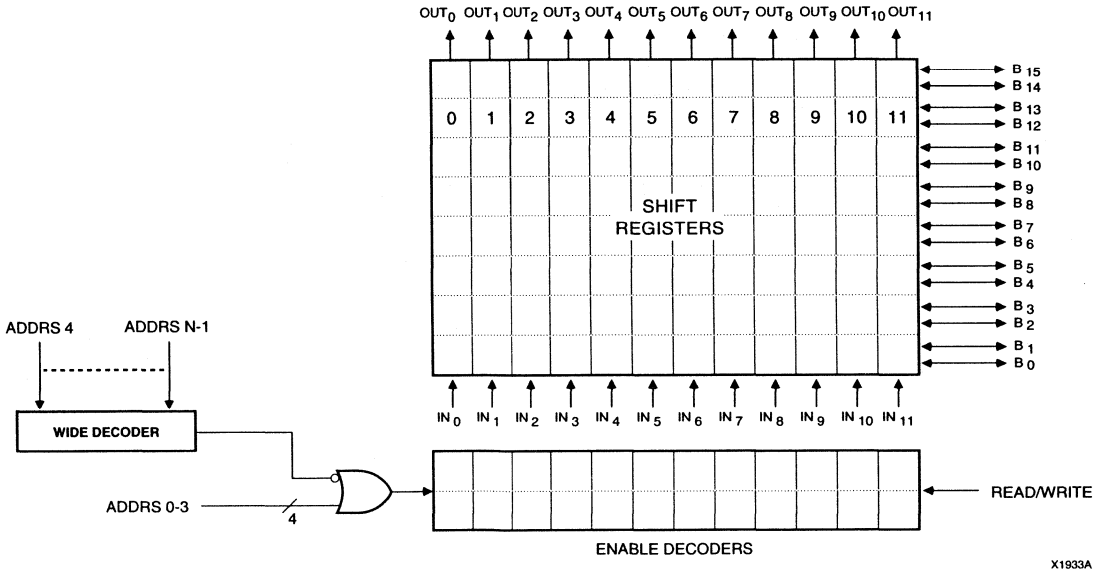


Figure 1. Serial Input/Output System

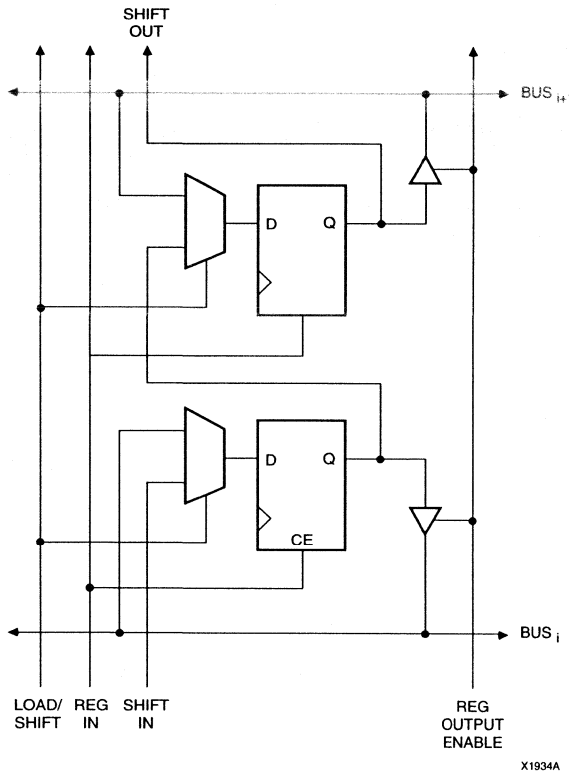


Figure 2. Shift Register CLB

select data. For loading, data is taken from the bus; for shifting, it is taken from the adjacent flip-flop. A register enable is also provided, that must be asserted for either loading or shifting the register.

The connections to the bus.are also shown in this diagram. A bidirectional bus has been chosen; both the inputs and outputs are connected to it. Alternatively, separate input and output busses could have been used. One Longline would broadcast data to the shift registers, and a second Longline would use the T-BUFs to multiplex the parallel outputs of the shift register. These busses could remain separate through the chip interface, or be combined into a single, bidirectional bus in the IOBs. Similarly, the shift-register inputs and outputs could remain separated, or combined for bidirectional operation.

Allowing space for control logic, 12 shift registers may be comfortably fitted onto the bus. These require a 4-bit bus address. This can be routed across the top of the shift registers and decoded at each column. A single CLB can decode the address, and use it to gate an enable signal. Two decoders are required for each shift register; one each for load enable and 3-state enable.

If these registers are part of a larger I/O register space, higher order address bits must also be decoded as chip select. Dedicated logic is provided along the edges of the chip to serve this exact purpose. Using these decoders is much faster than using CLBs, and they are free, because no CLBs are used.

In the decoder, the address bits from the IOBs are input to a wired AND. The inputs to this wired AND can be configured to be inverting or non-inverting. In this way, any fixed combination of ones and zeros can be detected. The XC4005 allows up to 28 address bits to be decoded in a single address decoder, and there are 16 such decoders.

While this totally synchronous I/O system is somewhat unrealistic, it does illustrate the use of the horizontal Longlines for bussing. If required, each shift register could have been clocked separately. This would necessitate the synchronization of the load enables to the individual clocks. However, only 120 of the 196 CLBs have been utilized, and ample space remains for this minor task and any other control functions.

While any combination of functions could be implemented and bussed together in this way, counters are particularly interesting. The dedicated carry logic embedded into each CLB allows loadable counters to be implemented with the same density as the shift registers; two bits per CLB. This would permit the construction of a 12-channel, 16-bit counter/timer.

Note: Implementing the extensive bus structure discussed in this Application Note requires considerable expertise in LCA design. The designer must specify the Longlines to be used, and constrain the placement of logic around them. The approach is only recommended for experienced LCA designers.

XCELL is a quarterly newsletter, sent free-of-charge to all active Xilinx users. The purpose is to inform our customers about device and software availability, about new technical developments, about design methodology, problems and work-arounds, about additional electrical parameters not covered in the Data Book, about clever circuits and tutorial topics, and about simple solutions to perplexing problems.

The idea is to bring up-to-date information to our customers and make it easier to design with Xilinx devices and development systems.

Please direct all inquiries, comments and submissions to:

Editor: Brad Fawcett

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: (408) 879-5097
FAX: (408) 879-4676
E-Mail: brad.fawcett@xilinx.com

XCELL The Best of

THE NEWSLETTER FOR XILINX PROGRAMMABLE LOGIC USERS

Issues 1 through 11 1989 to 1993

Making Fabless Strategies Work

Xilinx is one of over 100 semiconductor companies that does not own its own wafer fabrication facility, and uses independent silicon foundries for fabrication services. Fabless companies are not a fad; their streamlined structure fits today's tumultuous, fast-moving market. Being fabless allows Xilinx to concentrate on what we do best - the design and marketing of programmable logic devices.

Hewlett-Packard's announcement that it is quitting the foundry business, and the recent troubles of a few fabless companies have led some industry pundits to once again question the viability of fabless semiconductor suppliers. (H.P. is *not* one of Xilinx's wafer sources.) We strongly believe that the oracles predicting the demise of the fabless semiconductor vendor are wrong. While fabless is not the best business model for everyone, Xilinx and many other fabless companies will continue to succeed by establishing a win-win business relationship with our foundry partners.

The first key to a successful fabless strategy is to employ standard fabrication processes that are compatible with a variety of foundries. Xilinx FPGAs and EPLDs are based on 'plain-vanilla' SRAM and EPROM technologies. This allows us to benefit automatically from the industry's latest process improvements and to establish multiple foundry sources for our products. Standard processes can be quickly applied to an existing fabrication line without significant equipment and engineering investments. Multiple foundry sources ensure adequate and continuous product availability in case of disasters. Competition between foundries, as well as ongoing process and product improvements, ensure that price projections are met. In contrast, fabless companies with specialized processes have fewer potential suppliers and less leverage in the foundry market. If a foundry agrees to a specialized process, prices inevitably will be higher due to the special attention needed to get and keep that process under control.

The relationship between a fabless semiconductor company and its foundries must be long term, based on mutual trust, and of benefit to both parties. As mentioned above, Xilinx benefits from being fabless by being able to concentrate on designing innovative products without having to dedicate resources to running a fabrication line. Of course, from a financial viewpoint, we also benefit by gaining access to advanced fabrication processes without the huge capital investment required to build our own fabrication facility. However, to remain competitive, we must continue to innovate and provide value through the continued development of new and better products.

Our foundry partners benefit by being able to diversify their manufacturing capacity over different equipment markets; through Xilinx, they have gained access to a significant new market segment without incurring the expense of product and market development. Foundries minimize demand volatility through this market diversification. Assuming a long-term relationship, the foundry can improve its competitiveness compared to other manufacturers.

Xilinx's foundries have gained a significant additional benefit - the ability to use FPGAs as process drivers; that is, the technology that is used to drive and verify process advancements. Historically, DRAMs were considered ideal process drivers. However, processes for DRAMs and logic products have diverged, and FPGAs have become the optimum process technology driver for several reasons. FPGAs use a standard CMOS process, are silicon-intensive, and require state-of-the-art feature size and defect density. Like DRAMs, FPGAs are repetitive in their architecture, thus lending themselves to easy defect analysis. SRAM-based FPGAs are reprogrammable, allowing for 100% fault testing. Our foundries have learned that by applying 10% to 20% of their capacity to FPGAs, they gain excellent rewards from process control diagnostics. The resulting improvements to their processes can be applied to their other CMOS product lines. (It should also be noted that Xilinx employs its own process experts, who work closely with our foundry partners in the development and implementation of process technology improvements.)

Thus, Xilinx can effectively drive process improvements through our working relationships with our foundry partners. But these relationships must be based on mutual benefits. In the future, as in the past, this will be a necessary ingredient for success.

Bernie Vonderschmitt
President, Xilinx Inc.

XC3000 Readback Clarified

The ability to read back configuration data, as well as data stored in flip-flops and latches, is crucial for the exhaustive device testing performed by Xilinx on every device before it leaves the factory.

Most of our customers have no need for this feature, but a few use Readback to verify that the configuration is still proper. This makes sense in applications that require uninterrupted operation, e.g. in telecom where the device may be configured once and then operates for months or years without ever being reconfigured.

To those few engineers who really need the readback feature, we apologize for the user-unfriendly interface and the sometimes sketchy documentation.

Here are some important considerations.

Use Readback only when necessary. Less than 1% of all LCA applications use it.

Readback does not interfere with normal LCA operation, but the flip-flop data being read back will be almost impossible to interpret unless the LCA device suspends its clocked operation during Readback.

Readback cannot be daisy chained. Even when the devices were configured in a daisy-chain, they must be read back individually.

Readback data comes out inverted, a configuration 1 becomes a readback 0, and vice versa.

Readback data contains variable flip-flop or latch data in most of the locations that were left unused during configuration. If you want to compare readback data against the configuration file, you must disregard (mask out) these locations as shown below.

Readback has no Preamble, and no second or third stop bit at the end of each frame.

The first frame starts with two dummy zeros instead of the single start bit (l) preceding every other frame. Remember, everything is inverted: Readback start bits are ones, stop bits are zeros.

Before the device is being configured, Readback must be enabled by the MakeBits menu.

0 means never,

1 means once, and

Cmd means on command.

Readback is initiated by a rising edge on M0. Rising edges on the CCLK input then clock out the Readback data, using the Ml pin as an output. The first rising edge of CCLK does nothing. The second and third rising edges clock out the two leading dummy zeros. The fourth and subsequent rising edges of CCLK clock out frame information, interspersed with a single O for stop at the end of each frame, followed by a single 1 for the start of the following frame. After the last frame stop bit has been clocked out, the Ml pin goes 3-state and further CCLK pulses are ignored.

Verifying Configuration Bitstream

In order to verify the integrity of the LCA configuration, you must compare the Readback bitstream against the configuration bit stream in all those positions not masked out by a 0 in the Mask bitstream.

Configuration bitstream and Mask bitstream have a common format, both are created from the MakeBits menu. Since the Readback bitstream format is different, as described above, you must adjust the formats before verification.

Either: Pad the Readback bitstream with preamble, two additional stop bits, and change the two dummy bits preceding the first frame to a normal start bit,

Or, better: Strip the Configuration and Mask bitstreams of the preamble, delete two of the three stop bits and create the two dummy bits at the beginning of the first frame. Always remember that Configuration and Readback have opposite polarity.

After the three bitstreams have been normalized you can perform the verification.

There is an error when (Readback = Configuration AND Mask = 1.

PA

For XC4000 Readback details, see the applications section of our 1993 Databook.

Park CCLK High

Remember that the CCLK pin of XC2000 and XC3000 devices must not be held Low for more than 5 µs. Dynamic circuitry inside the LCA can reach an unknown state when capacitors lose charge during excessive CCLK Low time, especially at high temperature, when leakage current is high.

If CCLK is held Low after configuration, a subsequent Readback may not function properly, reading back wrong information. Make sure that CCLK has been parked High for several milliseconds before the beginning of Readback.

TCW

The Secret of "Tie"

Before generating the configuration bit stream, the user has the option to tie or not tie the design. To "tie" means to create additional interconnects that terminate all floating transistor inputs or metal interconnects to well-defined levels or signals.

In a tied design, all inputs and interconnects are always High or Low, or are connected to a signal that switches between defined levels. In a non-tied design, the unused inputs and pieces of interconnect (there usually are more unused ones than used ones) are left floating. This poses no first-order problem, since these inputs are really not used. In this respect, it differs from the well-known problem of floating TTL inputs that are supposed to generate a High level.

The problem with undefined input levels in CMOS logic is that they may drift to the midpoint between Vcc and ground, half turning on both the pull-down and pull-up transistor, making a CMOS gate draw measurable Icc. Also, such undefined inputs may be affected by crosstalk from adjacent lines, thus increasing dynamic power consumption.

An untied design is likely to have increased dc and ac power consumption and increased on-chip noise. That's a good enough reason to spend the extra effort to tie every design.

To tie a design, select the -T option in the XDM MakeBits menu.

The Tilde De-Mystified

Timing values given by XACT or APR are sometimes preceded by the symbol ~, called Tilde by its Spanish name, but really meaning "approximately". How should the user interpret this symbol?

All non-tilde timing values given by XACT or APR are carefully simulated, modeled, and measured worst-case values, guaranteed over the range of processing tolerances and temperature and supply-voltage variations. The user can have confidence that no device will ever exceed these values.

The tilde is a disclaimer. It means that the delay is generated by so many concatenated resistor (or pass-transistor) -capacitor elements, that our design and test engineers have less confidence in the accuracy of the model and the repeatability of the timing value. Xilinx cannot guarantee it as an absolute worst-case value.

The number following the tilde is still a conservative specification; most likely the parameter in question is better than this value. But there is not the same guarantee as there is with non-tilde values. What is the user to do?

If a "tilde-value" is critical to your design, you have two choices:

1. Change the lay-out or routing such that the long uncertain delay is broken up into two "non-tilde" values, either by passing the net through a BIDI or through an unused CLB, or by dividing the net into two branches.
2. Add 25% to the value and ignore the tilde, making the reasonable assumption that this factor 1.25 compensates for the modeling uncertainty.

XC2064 and XC2018 ACLK delay values, though below 10 ns, are sometimes preceded by a tilde. You can safely ignore the tilde in these cases.

There has been a misleading explanation that the tilde indicates propagation delay differences between the rising and the falling edge of a signal. This is not true. Different from original 2.0µ technology XC2000 parts, all newer technology devices, and especially the XC3000 family parts, have their delays finely balanced.

Our designers have painstakingly adjusted n-channel and p-channel geometries to achieve driving impedances and threshold voltages that guarantee virtually identical propagation delays for rising and falling transitions.

Maybe we have been overly pessimistic and caused unjustified concern with the tilde. But we prefer to be cautious and make a distinction between worst-case guaranteed values and intelligent, albeit conservative, estimates.

PA

Configuring Devices in Parallel

In the special case where several LCA devices contain identical configuration data, they can be configured simultaneously to reduce program size and configuration time. When the program is stored in a Serial PROM, just make one LCA device the Master, all others the Slave, interconnect all CCLKs, and drive all DINs in parallel from the Serial PROM.

There are no timing problems. Between the 666 ns cycle time and the 400 ns access plus 60 ns set-up time, there are over 200 ns available for additional delay. This accommodates at least 250 pF of additional capacitive loading. The 1 MHz max specification in the Data Book only applies to READBACK; during configuration CCLK can be up to 1.5 MHz.

IOB Options

Our Data Book describes the operation of the XC3000 IOBs and their configuration options. This description is not complete: The activation of the passive pull-up is really coupled with the Three-State Enable, giving the following four choices:

- Passive pull-up activated, output buffer permanently three-stated (pin is input only, with pull-up).
- Passive pull-up de-activated, output buffer permanently three-stated (pin is input only, no pull-up).
- Passive pull-up de-activated, output buffer active, i.e. three-state control permanently de-activated (pin is ouput only).
- Passive pull-up de-activated, output buffer controlled by three-state control signal (pin can be I/O).

In other words:

The passive pull-up can only be used on pure inputs, not on I/O pins. The three-state control logic can be permanently disabled, resulting in a permanently active output. The other four options.

OUT INVERT,
THREE-STATE INVERT
OUTPUT SELECT
SLEW RATE

are not interdependent; they operate as described.

The XC4000 output pull-up and pull-down resistors do not have this limitation. They can be used with active outputs.

Unused Pins

Xilinx Programmable Gate Arrays come with an abundance of user I/O pins, from 58 on the XC2064 to 144 on the XC3090. Many applications leave a few, or even many, of these pins unused, but even unused pins need some attention.

Modern CMOS devices have extremely low input-leakage current, perhaps only a few nanoamps. (The 10µA guaranteed specification represents a testing limitation, not a real input current.)

Left disconnected, such an input could therefore float to any voltage. Clamp diodes prevent excursions above the supply voltage and below ground, thus protecting the input gate from destructive breakdown voltages. This leaves the problem of inputs floating uncontrolled between Vcc and ground.

An input voltage close to the threshold value 1.2V for TTL level-compatibility, 2.5V for CMOS level-compatibility will turn the input buffer partially on, thus creating a static current path from Vcc to ground and causing static power dissipation. Such a biased buffer also acts as a fairly high gain amplifier, making the circuit very susceptible to noise, crosstalk, ground-bounce and other undesirable disturbances.

It is, therefore, advisable to force unused inputs to a proper logic level.

XC2064 and XC2018

1. Leave unconfigured; externally connect to a High or Low level.
2. Configure as active output driven by an internally defined signal.

XC3000 and XC4000

Same as above, or

3. Configure as input with internal passive pull-up.

Putting unused I/O to use

An unused XC3000 series IOB can be used as part of the on-chip logic, e.g. as a shift register. Note that the associated package pin must be left free, and that the speed is not as high as it is with internal flip-flops.

Multiple I/O pins can also be used to perform the "wired AND" function in conjunction with an external pull-up resistor.

Don't Overshoot or Undershoot

Our 1992 Data Book explicitly forbids input voltage excursions more than 0.5 V outside the supply voltages (below ground, above Vcc). Hardly anybody would try to violate this with a static voltage or current, but many designs show PC-board reflections that sometimes exceed these rather tight limits. A better explanation of the problem is therefore in order.

All CMOS I/O pins are clamped against Vcc and against ground through diodes formed by the respective output transistors. Pure inputs have equivalent protection diodes. These diodes prevent any excessive voltage on the gate of the associated input transistor. Without such protection the input gate might accidentally get charged to a voltage that can rupture the gate oxide and thus destroy the input transistor. All modern MOS devices have such input protection.

What happens when the input voltage exceeds the specified limits?

Below -0.5 V, the ground clamp diode will start conducting, above Vcc + 0.5 V the Vcc clamp diode will start conducting. These diodes are fairly big and will clamp hundreds of milliamps with a voltage drop of less than 2 V. The problem is that this clamp current can stray into an area of the circuit where it might upset the internal logic. There is no hard data to quantify this concern, but our circuit designers feel uncomfortable about undefined currents of long duration in parts of the circuit that were not designed for that purpose.

Very high clamp currents (more than 100 mA at elevated temperature, more than 300 mA at room temperature) lasting for milliseconds can cause the parasitic bipolar input transistors to be triggered like an SCR, which then conducts unlimited Icc and thus destroys the device. Xilinx devices are extremely resistant to this latch-up.

Conclusion

Try to limit overshoot and undershoot to 0.5 V, the data sheet limit. If these values are exceeded, the clamp diodes will protect the inputs and limit the voltage swing. Large clamp currents of millisecond duration must be avoided at all costs, e.g. by adding current limiting series resistors.

Never drive inputs with active levels above Vcc, even when the Vcc supply is turned off. Strange things might happen during turn-on.

Function Generator Avoids Glitches

The combinatorial logic in all CLBs is implemented as a function generator in the form of a multiplexer, built out of transfer gates. The logic inputs form the select inputs to this multiplexer, while the configuration bits drive the data inputs to the multiplexer.

The Xilinx circuit designers were very careful to achieve a balanced design with similar (almost equal) propagation delays from the various select inputs to the data output.

The delay from the data inputs to the output is, of course, immaterial, since the data inputs do not change dynamically. They are only affected by configuration.

This balanced design minimizes the duration of possible decoding glitches when more than one select input changes. Note that there can never be a decoding glitch when only one select input changes. Even a non-overlapping decoder cannot generate a glitch problem, since the node capacitance will retain the previous logic level until the new transfer gate is activated about a nanosecond later.

When more than one input changes "simultaneously," the user should analyze the logic output for any possible intermediate code. If any such code permutation produces a different result, the user must assume that such a glitch might occur and must make the system design immune to it. The glitch might be only a few nanoseconds long, but that is long enough to upset an asynchronous design.

If none of the possible address sequences produces a different result, the user can be sure that there will be no glitch.

The designer of synchronous systems generally doesn't worry about such glitches, since synchronous designs are fundamentally immune to glitches on all signals except clocks or direct SET/RESET inputs.

Worst-Case Input Set-Up Time

Timing parameters in programmable devices are more difficult to specify than in fixed-program devices, because the user can affect some parameters through routing.

Inside the LCA, a synchronous design is easy to analyze, because hold time is not an issue, since clock skew is much shorter than the minimum clock-to-Q delay of any CLB. The only concern is for performance: Is the sum of propagation delay and set-up time less than the clock period?

The set-up time at the LCA input is more complex, since the clock delay from the clock pad to the internal clock cannot be ignored.

The data sheet specifies the IOB set-up time with respect to its clock (not with respect to the clock pad!). The unavoidable delay from clock pad to internal clock must obviously be subtracted from the specified set-up time, to arrive at the system set-up time.

What is the maximum value for the input set-up time, and what is its minimum value? Is there a risk for a hold-time requirement?

Maximum Set-up Time

The longest input pad set-up time, the one that determines system performance, is the specified longest IOB flip-flop set-up time minus the shortest clock delay that is consistent with such a long setup time.

The question is:
How well do such delays track

Here is one unrealistic assumption:

"All delays track perfectly. In a given part, at any given temperature and supply voltage, the ratio of any actual parameter value to its specified worst-case value is the same constant."

If this were true, the max set-up time would simply be the difference of the two specified max values for flip-flop set-up time and clock delay.

Here is another unrealistic assumption: "There is no delay tracking. Any parameter can vary between its max and min value, independent of all other parameters."

If this were true, the max system set-up time would be the difference between the specified max flip-flop set-up time and the minimum clock delay, whatever small value that might be.

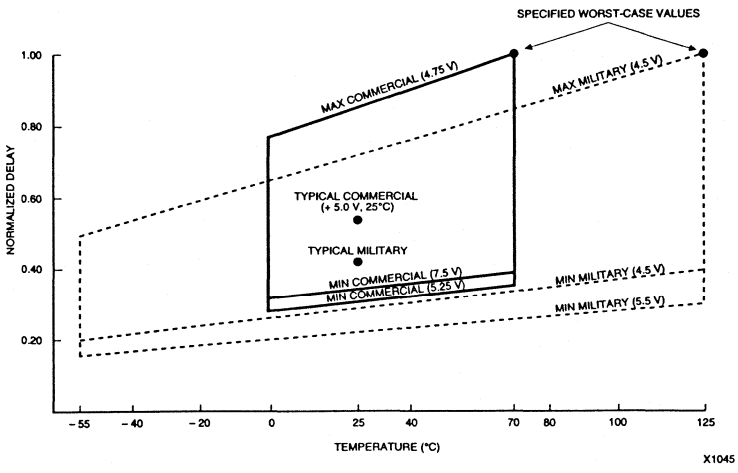
Both these assumptions are wrong.

The circuits being evaluated reside on one piece of silicon. They were processed together, and they have a common temperature and supply voltage. All delay parameters will, therefore, track reasonably well. But since all parameters do not necessarily depend on the same physical phenomena (resistance, capacitance, threshold voltage etc.) in the same way, they will not track perfectly.

We make the assumption that tracking in any one device will be better than 70%.

All ratios of actual delay to specified worst-case delay for all parameters on the same device at any instant will be within a two-to-one range.

- If one delay is close to the specified max value, then all the others will be between 70% and 100% of their respective max values.
- If the relatively slowest parameter is at 50% of its specified max value, then all the other parameters will be between 35% and 70% of their respective max values, etc. (The user should feel safe with this conservative assumption. In reality, parameters track much better than this.)



The longest set-up time is, therefore, the specified max IOB flip-flop set-up time minus 70% of the specified max clock delay.

Example:
XC3020-100 using CMOS compatible TCLK input:
Ts max= 17 ns - 0.7 • 7 ns = 12.1 ns

Minimum Set-Up Time and Possible Hold Time Requirement

The shortest possible set-up time is the minimum IOB set-up time minus the longest value for the clock delay that is consistent with such a short set-up time.

The minimum value for the flip-flop set-up time is not specified, since it is not readily testable. A very conservative guess puts it as short as 10% of the specified max. value. This can only occur at low temperature and high Vcc.

In line with the previous discussion about tracking, the maximum clock delay might then be as long as 14% of its specified max value.

Example:
XC3020-100 using CMOS compatible TCLK input:
Tsmin=0.1 • 17ns - 0.14 • 7ns=0.7ns

This means that the data-to-clock set-up time window on the LCA inputs (pads) is always somewhere between 12.1 ns and 0.7 ns. This is a wide range, but the value is always positive.

There is no hold time requirement. Data may change simultaneously with the clock, provided the clock drives the CMOS-compatible LCA input and uses the Global or Alternate clock distribution network.

PA

Set-Up and Hold Times

Set-up time describes the requirement for valid input data prior to the clock edge.

Hold time describes the requirement for valid input data after the active clock edge.

Any particular flip-flop at a particular temperature and supply voltage clocks in the data that happens to be at its input during an extremely narrow picosecond timing window. (If data changes during this narrow window, the flip-flop goes metastable). The width of this window is constant, but its position varies, depending on processing, temperature and Vcc.

The longest set-up time describes the earliest possible position for this window; the longest hold time describes its latest possible position. If no hold time is specified, the set-up time will always be positive, i.e. the window will always be before the clock edge.

These critical set-up and hold time values are often listed in the min column of the data sheet, conforming to an ill-conceived convention established in early 7400 data sheets.

Enlightened people have argued for decades that these are really max limits of device parameters, but it has become senseless to fight over form when (hopefully) everybody agrees on the meaning.

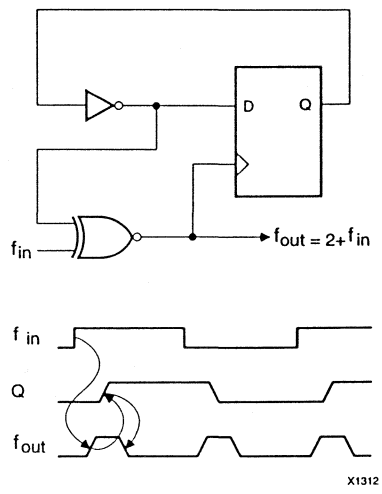
Double the Clock Frequency

A 50% duty cycle input can be doubled in frequency, provided the resulting 2-f clock can tolerate a wide variation in duty cycle. The circuit below generates an output pulse in response to each transition on the input.

The output rising edge is delayed one T_{ILO} from either input transition. The output High time is the sum of a clock-to-Q delay plus two T_{ILO} delays, about 25 ns in a fast part. This output pulse will clock other flip-flops on the same die reliably. (At low temperature and high Vcc the pulse will be shorter, but the flip-flop response is also faster under these conditions.)

This asynchronous circuit is frowned upon by all true digital designers. It should only be used as a tool of last resort.

PA



Rube Goldberg and the Art of LCA Design

Xilinx FPGA devices bring gate-array capability to the large number of logic designers who cannot afford the cost, risk, and delay of a masked gate array, but still want to design their own LSI circuits. More new gate array designs are presently being implemented with Xilinx FPGAs than with all other gate-array technologies combined. We love this wide acceptance of our technology, but we are also concerned about the sometimes marginal and even bad logic designs that are being implemented in our parts.

There are exciting arguments that make Xilinx FPGAs your favorite choice:

"Build your design in parts, try it out, and modify it if it doesn't work!"
"Fix your mistake before your boss ever sees it!"
"Respond easily to demands created by the market or your competition!"
"Convert an idea you had in the shower to a working chip the same day!"

These are perfectly valid statements, but they do not guarantee design quality. In fact they might actually tempt the designer to be less thorough in the original approach. "Why perform a careful analysis when I can try it out so easily?"

This attitude can lead to bad design methods and unreliable products. Here are some words of advice, based on over 30 years of systems and logic design experience, and exposure to a few hundred LCA designs over the past two years:

- Always start off with a top down design. Look at the big picture before you implement the details. Draw a block diagram, step back, and see if you can simplify it by combining functions.
- Trade off complexity against speed. LCA circuitry is quite fast. If your clock runs much slower than 10 MHz, investigate whether you might perform the function time-multiplexed or serially.
- When you design slow logic, don't get careless. The circuitry doesn't know about your low speed; it can still react to nanosecond decoding spikes on the clock or asynchronous reset lines, and might be sensitive to 10ns clock skew problems.
- Be extremely careful with asynchronous inputs. Whenever two asynchronous signals are combined, always (ALWAYS!) perform a thorough worst-case analysis of what happens under the most extreme phase relationships between those inputs. Use the combination of two asynchronous signals to affect only one flip-flop, either to clock it, to reset it, or to synchronize the two signals. One flip-flop will either react or not react to a marginal signal; but several flip-flops might disagree and may cause your system to crash. Remember, according to Murphy's Law, if something bad can happen, it will happen, and usually at the most inappropriate moment.
- Never use a decoder to clock or reset a flip-flop or latch asynchronously. Uncontrollable decoding spikes may cause unpredictable behavior that may be temperature-, voltage- or lot dependent. A classical example: Using the Terminal Count output from a 74161 as a clock is an invitation to disaster.
- Be aware of the metastability problem, but don't be paranoid about it. Read about it in the Application section of our Data Book. It shows that an extra 10 ns of tolerable propagation delay can reduce the metastability problem to statistical insignificance.
- Use the global clock distribution network which eliminates all clock skew problems. If you have to distribute a clock signal through general-purpose interconnect and "magic boxes" to several flip flops, always check for clock-skew problems, even if your design is otherwise not time-critical. There usually is an easy cure: Run the clock delays in a direction opposite to the data flow. Clock the most significant part of a counter early, the less significant part later, and all clock skew problems disappear.
- If you have used non synchronous logic tricks, analyze them very carefully. Check for potential problems with faster parts. Evaluate your design in the fastest LCA that you can buy (presently the -100), to check for potential future problems. There are also two simple ways to change the delay in LCAs mounted on your board: Just vary the temperature or the supply voltage: Heat

makes CMOS slow, cold makes it fast; low Vcc makes CMOS slow, high Vcc makes it fast.

If your design fails at high temperature or low Vcc, then you are just pushing performance and are running into problems with excessive propagation delays. You might want to improve the routing or use a faster part.

If, however, your design fails at low temperature and / or high Vcc, you have reasons to worry. Take a deep breath and analyze your design for asynchronous abnormalities and for clock skew problems. If you don't fix these problems immediately, they will bite you in the future. CMOS processes are constantly being improved and shrunk. Circuits will get faster, and what was an innocent glitch in a slow part may jeopardize your system in the future.

Logic design is both an art and a science. There are elegant designs and there are kludges; there are rugged designs and there are flimsy contraptions that will inevitably fail sooner or later.

Standard LSI circuits are crafted by experienced logic designers who know what's at stake, are aware of the possible pitfalls and know how to avoid them. And, they simulate their designs and take the time to get it right.

Programmable gate arrays give the user full responsibility for every aspect of the logic design. We hope that the user community is up to this challenge.

The cartoonist Reuben Lucius Goldberg (1883-1970) was known for his whimsical drawings of ludicrously intricate machinery meant to perform simple tasks.

PA

The Effect of Marginal Supply Voltage

Since Xilinx LCA devices store their configuration in static latches, some users have asked about the integrity of the configuration program under abnormal supply voltage conditions.

Here is a complete description of XC3000 and XC4000 device behavior during supply ramp-up and ramp-down.

When Vcc is first applied and is still below about 3 V, the device wakes up in the pre-initialization mode. HDC is High; $\overline{\text{INIT}}$, $\overline{\text{LDC}}$ and D/$\overline{\text{P}}$ are Low, and all other outputs are 3-stated with a weak pull-up resistor.

When Vcc has risen to a value above ~3 V, and a 1 and a 0 have been successfully written into two special cells in the configuration memory, the initialization power on time delay is started. This delay compensates for differences in Vcc detect threshold and internal CCLK oscillator frequency between different devices in a daisy chain. The initialization delay counts clock periods of an on-chip oscillator (CCLK) which has a 3:1 frequency range depending on processing, voltage and temperature. Time-out, therefore, takes between 11 and 33 ms for a slave device, four times longer for a master device.

This factor of four makes sure that even the fastest master will always take longer than any slave. We assume that the worst-case difference between 33 ms and 4 x 11 ms is enough to compensate for the Vcc rise time spent between threshold differences (max 2 V) of devices in a daisy chain. Only in cases of very slow Vcc rise time (>25 ms), must the user hold $\overline{\text{RESET}}$ Low until Vcc has reached a proper level.

After the end of the initialization time-out, each device clears its configuration memory in a fraction of a millisecond, then tests for inactive $\overline{\text{RESET}}$, stores the MODE inputs and starts the configuration process, as described in the DataBook. After the device is configured, Vcc may dip to about 3.5 V without any significant consequences beyond an increase in delays (circuit speed is proportional to Vcc), and a reduction in output drive. If Vcc drops into the 3-V range, it triggers a sensor that forces the device back to the pre-initialization mode described above. All flip-flops are reset, HDC goes High; $\overline{\text{INIT}}$, $\overline{\text{LDC}}$ and D/$\overline{\text{P}}$ go Low, and all other outputs are 3-stated with a weak resistive pull-up. If Vcc dips substantially lower, the active outputs become weaker, but the device stays in this pre-initialization mode. When Vcc rises again, a normal configuration process is initiated, as described above.

The user need not be concerned about power supply dips: The XC3000/XC4000 device stays configured for small dips, and is "smart enough" to reconfigure itself (if it is a master) or to ask for reconfiguration by pulling $\overline{\text{INIT}}$ and D/$\overline{\text{P}}$ Low (if it is a slave). The device will not lock up; the user can initiate re-configuration at any time just by pulling D/$\overline{\text{P}}$ Low or, if D/$\overline{\text{P}}$ is Low, by forcing a High-to-Low transition on $\overline{\text{RESET}}$.

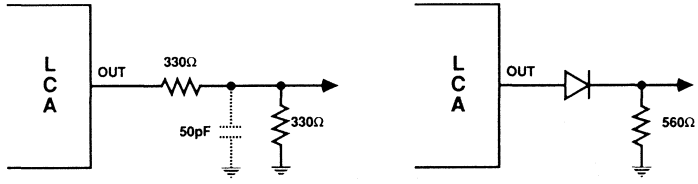
Ground Bounce

Activating or changing a large number of output pins simultaneously can lead to voltage spikes on the ground and Vcc levels inside the chip. The output current causes a voltage drop in the supply distribution metalization on the chip, in the bonding wires and the lead frame. Worse is the inductive voltage drop caused by the current change over the bonding wire inductance.

This is a well-known problem not only with fast bipolar or CMOS interface devices, but also with high pin-count gate arrays. It is commonly referred to as "ground bounce", because the change in ground potential is more critical than the equivalent change in Vcc potential. (TTL-oriented systems have far less noise immunity at the Low level than at the High level).

Xilinx circuit designers have given the LCA devices a very good Vcc and ground distribution metal grid on the chip, as well as double bonding to every supply pin. Packages below 100 pins have two Vcc and ground pin pairs, packages above 100 pins have eight Vcc and ground pin pairs to reduce supply lead resistance and inductance. What can the user do to minimize ground bounce?

- Provide solid Vcc and ground levels. Use multi-layer boards and decoupling. *Wire-wrapping the supply connections is an invitation to disaster.*
- Absolutely always connect all Vcc and ground pins.
- Configure outputs XC3000 slew-limited whenever the required performance allows this. This is the default option. Slew-limited outputs reduce transient amplitude by 75%.
- Use CMOS input levels whenever possible. This increases input noise immunity from less than 1 V to over 2 V.
- Stagger the activation or the change of output drivers by deliberately introduced unequal routing delays.
- Move trouble-causing outputs close to a package ground pin in order to minimize the device internal voltage drop. Move sensitive inputs, like clocks, close to another package ground pin.
- Finally, if there still is a ground bounce problem on a few outputs, attenuate and/or filter these outputs. A 50% attenuater (330Ω, 330Ω) perhaps combined with a 50 pF decoupling of the center point will reduce V_{OL} and calm it down. Changing the upper resistor to a diode might improve the situation even more.



Three-State vs Output Enable

The control input that causes an IOB output or Longline driver to go into the high impedance state is called (active High) "Three-State" in Xilinx literature and in XACT. The same signal is commonly known as (active Low) Output Enable or $\overline{OE}$.

These two signals are identical, i.e. T=$\overline{OE}$, as explicitly stated in our Data Book.

To put it more bluntly: T is not an active High Output Enable, rather it is identical with an active Low OE.

"Tri-state" is a registered trademark of National Semiconductor who pioneered this concept on TTL outputs in the late sixties. The names "Three-state" or "3-State" are ways around this trademark. The name refers to the third state of an output, beyond active High and active Low.

Powerdown Operation

A Low level on the $\overline{\text{PWRDWN}}$ input, while Vcc remains higher than 3 V, stops all internal activity, thus reducing Icc to a very low level:

- All internal pull-ups (on Long lines as well as on the I/O pads) are turned off.
- All package outputs are three stated.
- All package inputs ignore the actual input level and present a 1 (High) to the internal logic.
- All internal flip-flops or latches are permanently reset.
- The internal configuration is retained.
- When $\overline{\text{PWRDWN}}$ is returned High, after Vcc is at its nominal value, the device returns to operation with the same sequence of buffer enable and D/$\overline{\text{P}}$ as at the completion of configuration.

Things to Remember

Powerdown retains the configuration, but loses all data stored in the device. Powerdown three-states all outputs and ignores all inputs. No clock signal will be recognized. Any input level between ground and the actual Vcc is allowed. All internal flip-flops and latches are permanently reset and all inputs are interpreted as High, but the internal combinatorial logic is fully functional.

Things to Watch Out for

Make sure that the combination of all inputs High and all internal Qs Low in your design will not generate internal oscillations or create permanent bus contention by activating internal bus drivers with conflicting data onto the same long line. These two situations are farfetched, but they are possible and will result in increased power consumption. It is quite easy to simulate these conditions since all inputs are stable and the internal logic is entirely combinatorial, unless latches have been made out of function generators. Make sure that no applied signal tries to pull any input more positive than the actual supply voltage (Vcc). This would feed Vcc through the input protection clamp diode.

Input Current is Zero

Some designers keep asking about input current. Let us state bluntly:

The input current is negligible, just nanoamps, if

- The output sharing the same pin is three-stated,
- The internal pull-up option is not activated,
- The device is not in configuration mode where many pins have internal pull-ups.
- Vcc is above 4V.
- $0 \le V_{IN} \le V_{cc}$

If you ever observe our inputs hogging the drive voltage, you must have done something wrong. Make sure you counted the pin number right—and in the right direction, that you configured the device properly, and that Vcc is up. Then use an oscilloscope and multimeter, but please don't use the phone. Our inputs don't draw any current worth talking about, typically < 100 nA!

PA

Don't Pre-Assign Package Pins

In theory, FPGAs offer the system designer the option to preassign the package pins and lay out the PC board before completing the detailed design of the IC.

This method works well when the FPGA is sparsely populated and, therefore, has the additional routing resources to accommodate an imposed pinout.

For typical designs this method does not work well. We have seen many cases where the FPGA could not be routed with the imposed pin-out, but could be routed once the pin-out was left free. This leads to daughter board unscramblers or to a relay out of the PC board, headaches and expenses that the user would like to avoid.

So, as a rule, wait with the pin-out assignment until the LCA device has been routed. The exception to this rule are very sparsely populated designs or designs with very limited I/O.

XC2000/XC3000 CCLK Low Time

Most of the circuitry in our devices is static, i.e. the chip will work down to zero clock frequency.

CCLK is the exception. Its circuitry is half-static, half-dynamic and does not tolerate a Low time in excess of 5 µs. For very low speed operation, you can stretch the CCLK High time to any desired value, but keep the Low time short.

XC4000 does not have this restriction.

Just Say NO to Asynchronous Design

Synchronous designs are safer than asynchronous designs, more predictable, easier to simulate and to debug. Asynchronous design methods may ruin your project, your career and your health, but some designers still insist on creating that seemingly simple, fast little asynchronous circuit.

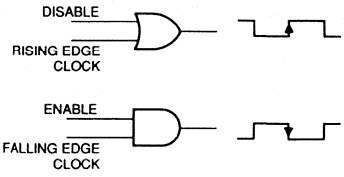
Twenty years ago, TTL-MSI circuits made synchronous design attractive and affordable; fifteen years ago, synchronous microprocessors took over many hardware designs; more recently, synchronous State Machines have become very popular, but some designers still feel the itch to play asynchronous tricks.

The recent popularity of ASICs has created a new flurry of asynchronous designs in a specially treacherous environment: Gate Arrays and Programmable Gate Arrays are being customized at the gate level, and may tempt the designer to develop bad asynchronous habits, especially dangerous since it is very difficult to inspect internal nodes, and impossible to calm them down with capacitive loads, the BandAid of simpler technologies.

Here is a short description of the ugly pitfalls in asynchronous design, documented for the benefit of the inexperienced designer. Veterans are familiar with the problems and may even know their way around them to design safe asynchronous circuits.

Clock Gating

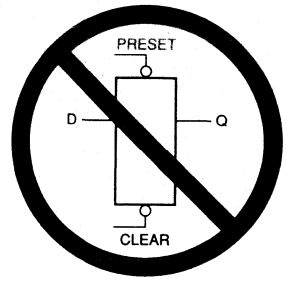
Gating a clock signal with an asynchronous enable or multiplex signal is an invitation to disaster. It will occasionally create clock pulses of marginal width, and will sometimes move the clock edge. A synchronous signal can be used to gate the clock reliably, as shown below, but this still introduces an additional clock delay, which can cause hold time problems.



Reliable Synchronous Clock Gating

Ripple Counters

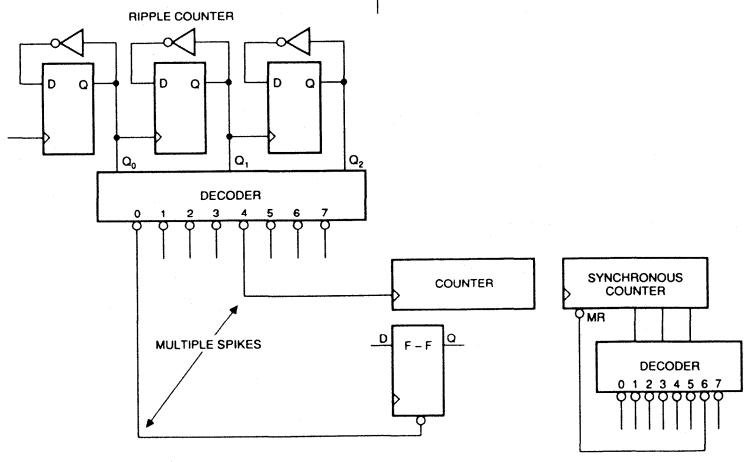
Using the output of one flip-flop to clock its neighbor can generate a binary counter of arbitrary length. The problem occurs when the counter increments from 2^n -1 to 2^n. It takes n delays from the incoming clock to the resulting change in bit n. In a 16-bit counter, this delay will be longer than 100 ns. At a 10 MHz clock rate, certain codes will never exist, the LSB will have changed before the MSB reached its new value. Decoding such a counter will produce dangerous decoding spikes. Note that these spikes are independent of the incoming clock rate. Designers of slow systems are actually most vulnerable to this problem, since they are less sensitive to delicate timing issues.



Decoder Driving Clocks and Reset Inputs

Indiscriminate use of decoder outputs to clock flip-flops or set/reset them asynchronously is one way to invite unpredictable and unreliable operation. The decoded outputs from synchronous counters are even more devious. While the decoding spikes from ripple counters are fairly wide and somewhat predictable, decoding spikes from synchronous counters are entirely the result of small but unpredictable differences in routing and decoding delays.

Using the decoded Terminal Count as asynchronous Master Reset input is another popular method to achieve unreliable operation. The spike might reset some flip-flops, but not all.



Unreliable Use of Decoders

Synchronizing One Input in Several Flip-Flops

A single asynchronous input should be synchronized in only one flip-flop. There will be an occasional extra metastable delay as described in the Applications section of our Data Book. This extra delay is acceptable in all but the very fastest systems. Synchronizing one input in more than one flip-flop is another matter. The set-up times and input routing delays of the various flip-flops will inevitably differ by one or several nanoseconds. Any input change occurring during this time difference will be clocked differently into the individual flip-flops, and the error will last for a full clock period. Synchronize any input with only one single flip-flop!

Synchronizing Multiple Inputs in One Register

Synchronizing an asynchronous parallel data word can lead to wrong results when the asynchronous inputs change during the register set-up time. For the duration of one clock period the register might then contain any imaginable mixture of old and new bit values. There is no simple solution, the most popular is to pipeline the result and compare the previous and present values. Any difference declares the data invalid. This operation is sometimes performed in software.

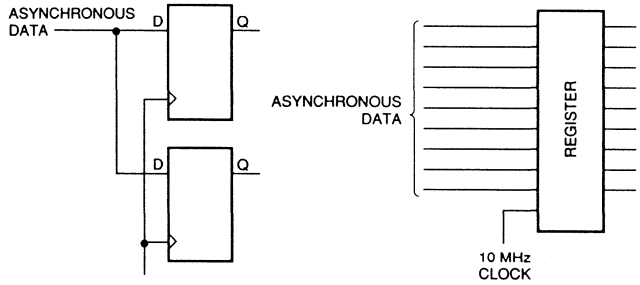
Asynchronous Reset of Multiple Circuits

A simple RC combination, perhaps augmented by a diode, is a popular power-on reset circuit. When it is used to drive several ICs in parallel, the system must accept wide variations in the reset duration. Differences in input threshold voltage will cause some circuits to start operating while others are still being held reset. If that is unacceptable, the RC combination must drive only one IC which, in turn, controls the reset operation of all others.

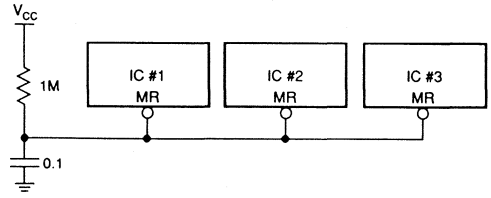
PA

Dot Your T's!

Schematic capture packages have an obsession about details. Some of them insist that a connecting dot be put on every T-joint, even on a connection to a bus. So, even if you think that it's redundant or ugly, put in the dots. It saves you from strange problems later on. One day in the future, we'll have true Artificial Intelligence, and computers will become our servants,not our masters. Until then, dot your T's!



Dangerous Methods of Synchronizing Asychronous Inputs



Asynchronous Reset of Multiple Circuits

Internal Bus Contention

The XC3000 and XC4000 families have internal 3-state bus drivers (TBUFs). As in any other bus design, such bus drivers must be enabled carefully in order to avoid, or at least minimize, bus contention. (Bus contention means that one driver tries to drive the bus High while a second driver tries to drive it Low).

Since the potential overlap of the enable signals is lay-out dependent, bus contention is the responsibility of the LCA user. We can only supply the following information:

While two internal buffers drive conflicting data, they create a current path of typically 6 mA. This current is tolerable, but should not last indefinitely, since it exceeds our (conservative) current density rules. A continuous contention could, after thousands of hours, lead to metal migration problems.

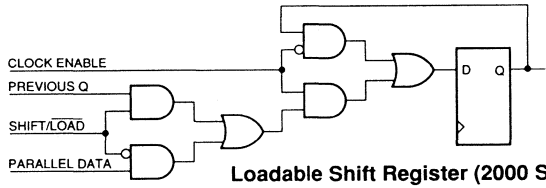
In a typical system, 10 ns of internal bus contention at 5 MHz would just result in a slight increase in Icc.

16 bits x 6 mA x 10 ns x 5 MHz
x 50% probability = 2.5 mA.

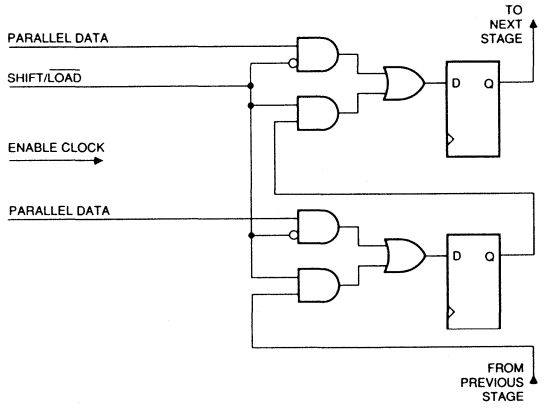
There is a special use of the 3-state control input: When it is directly driven by the same signal that drives the data input of the buffer, (i.e. when D and T are effectively tied together, the 3-state buffer becomes an "open collector" driver. Multiple drivers of this type can be used to implement the "wired-AND" function, using resistive pull-up.

In this situation there cannot be any contention, since the 3-state control input is designed to be slow in activating and fast in deactivating the driver.

Connecting D to ground is an obvious alternative, but may be more difficult to route. PA



Loadable Shift Register (2000 Series)



Loadable Shift Register with Clock Enable

The 2000 Series CLB primitive shown below is a building block for a shift register with synchronous load and clock enable, or for a bidirectional shift register with clock enable but without parallel load. The 3000 Series CLB primitive shown below is a 2-bit building block for a shift register with synchronous load and clock enable, or for a bidirectional shift register with clock enable but without parallel load.

Design Security

Some Xilinx customers are concerned about design security. How can they prevent their designs from being copied or reverse-engineered?

We must distinguish between two very different situations:

1. The design contains the configuration data in a serial or parallel EPROM or in a microprocessor's memory. This is the normal case.
2. The design does not permanently store a source of configuration, data. After the LCA was configured, the EPROM or other source was removed from the system, and configuration is kept alive in the LCA through battery-back-up.

1. In the first case, it is obviously very easy to make an identical copy of the design by copying the configuration data, the devices, and their interconnect patterns. Deleting the part-identifying markings on the top of the ICs would make the copying slightly more difficult, but the main defense is legal. The bitstream is easily protected by copyright laws that have proven to be more successfully enforced than the intellectual property rights of circuit designs.

   While it is easy to make an identical copy of the design, (clearly violating the copyright) it is virtually impossible to use the bitstream in order to understand the design or make modifications to it. Xilinx keeps the interpretation of the bitstream a closely guarded secret. Reverse-engineering an LCA would require an enormously tedious analysis of each individual configuration bit, which would still only generate an XACT view of the LCA, not a usable schematic.

   The combination of copyright protection and the almost unsurmountable difficulty of creating a design variation for the intended function provides good LCA design security. The recent successes of small companies in reverse-engineering microprocessor support circuits show that a non-programmed device can actually be more vulnerable than an LCA.
2. If the design does not contain the source of configuration data, but relies on battery-back-up of the LCA configuration, then there is no conceivable way of copying this design. Opening up the package and probing thousands of latches in undocumented positions to read out their data without ever disturbing the configuration is impossible.

   This mode of operation offers the ultimate design security.

PA

Nanowatts, Not Microwatts

LCA power consumption in the powerdown state has been somewhat of a mystery. The data book hints at nanowatts, but the published specifications only guarantee milliwatts.

We tested a representative sample of parts and found the powerdown current at room temperature and 5 V mostly below 50 nanoamps. This value is reduced in half at 2.5 V, but doubles for every 10 °C increase in temperature.

This is good news for battery-back-up. Even the tiniest lithium battery can power an LCA device for years.

Why don't we update our guaranteed specification? One reason is the difficulty of measuring very small currents on a high-speed production tester. Another one is the potential yield loss when this parameter happens to be higher. No reason to scrap a part for a parameter that only a few users are interested in.

PA

Powerdown Pin Must Be High For Configuration

A Low on the $\overline{\text{PWRDWN}}$ pin puts the LCA device to sleep with a very low power consumption, typically less than one microwatt. The on-chip oscillator is stopped, and the low-Vcc detector is disabled. During configuration, the $\overline{\text{PWRDWN}}$ pin must be High, since configuration uses the internal oscillator. Whenever Vcc goes below 4 V, $\overline{\text{PWRDWN}}$ must already be Low in order to prevent automatic reconfiguration at low Vcc. For the same reason, Vcc must first be restored to 4 V or more, before $\overline{\text{PWRDWN}}$ can be made High.

PA

Magnitude Comparator: Small, Fast, Expandable

A Magnitude comparator is more complex than an identity comparator, but simpler than an adder or subtracter. A magnitude comparator indicates not only when two operands are equal, but also which one is greater if they are unequal. PA

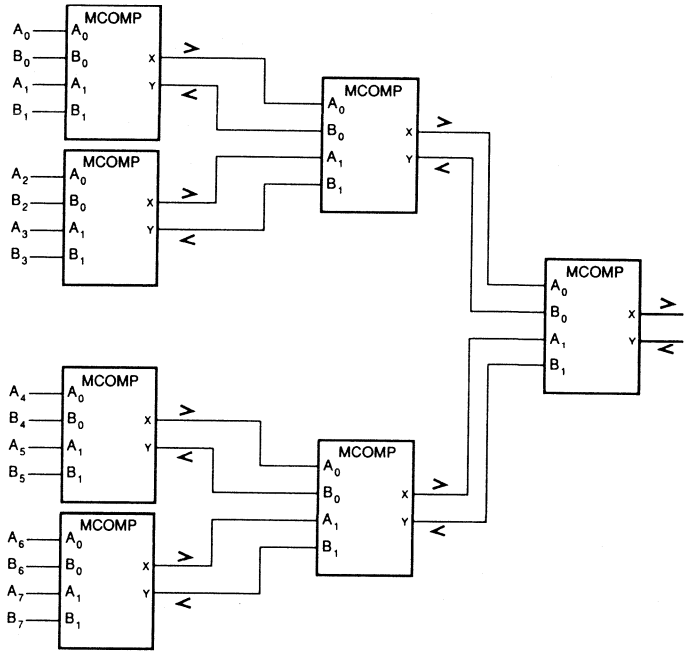
Truth Table

Bl	Al	B0	A0	A>B	A<B
0	0	0	0	0	0
0	0	0	1	1	0
0	0	1	0	0	1
0	0	1	1	0	0
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	0	1
1	0	0	1	0	1
1	0	1	0	0	1
1	0	1	1	0	1
1	1	0	0	0	0
1	1	0	1	1	0
1	1	1	0	0	1
1	1	1	1	0	0

This truth table is represented by the following equations:

$A>B: = A_0 \bullet \overline{B}_0 \bullet (A_1 \text{ XOR } \overline{B}_1) + A_1 \bullet \overline{B}_1$

$A<B: = \overline{A}_0 \bullet B_0 \bullet (A_1 \text{ XOR } \overline{B}_1) + \overline{A}_1 \bullet B_1$



Magnitude Comparator Expands to Any Size

LCA Drives Liquid Crystal Display Directly

Non-multiplexed Liquid Crystal Displays (LCDs) must be driven with an ac voltage of 30 to 100 Hz and 10 V peak-to-peak amplitude, without any DC component.

Generating this voltage is surprisingly simple in an LCA, using only half a CLB plus one IOB per segment. The back plane of the display is driven by a low frequency (100 Hz) square wave BP, oscillating between 0 and +5 V, and this signal is also used to control the inverting/non-inverting of Data.

When DOUT is in phase with BP, there is no ac-voltage across the segment, and it looks transparent. When DOUT is in counter-phase with BP, there is an ac-voltage across the segment, and it appears dark=on.

An additional Light Blanking Input (LBI) can force data to be blank=zero, useful for leading-zero suppression. NJC

Comparing Data on a Bus

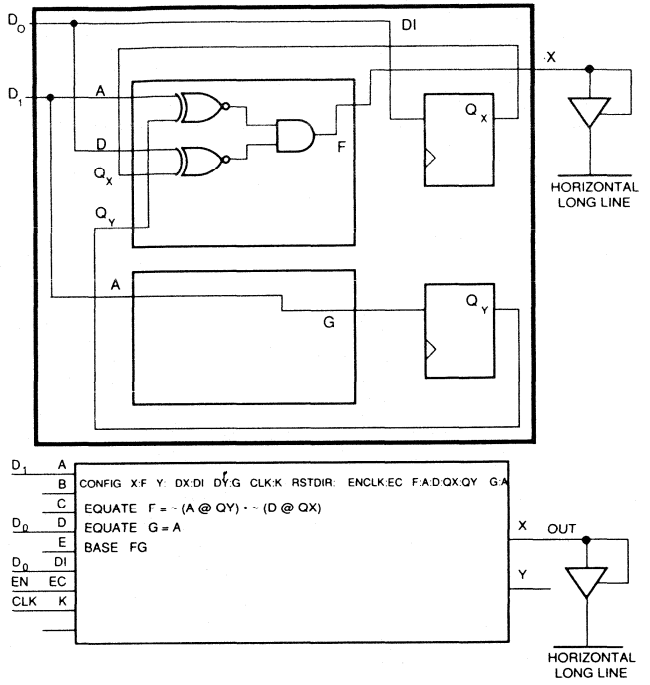
Some systems need to compare variable data on a bus against a value that had previously been loaded from the same bus. Such an identity comparator can store and compare two data bits per CLB, then using a Long Line to AND the result.

When Enable Clock is active, D0 (through .di) is clocked into Qx, while Dl (through .a) is clocked into Qy. D0 is also routed to the .d input.

The F function generator is brought out and drives the T input of a Longline buffer. F is High when the two incoming bits match the registered bits.

F = (A XOR Qy) • (D XOR Qx)

PA



Very Fast Accumulator with Pipelined Carry

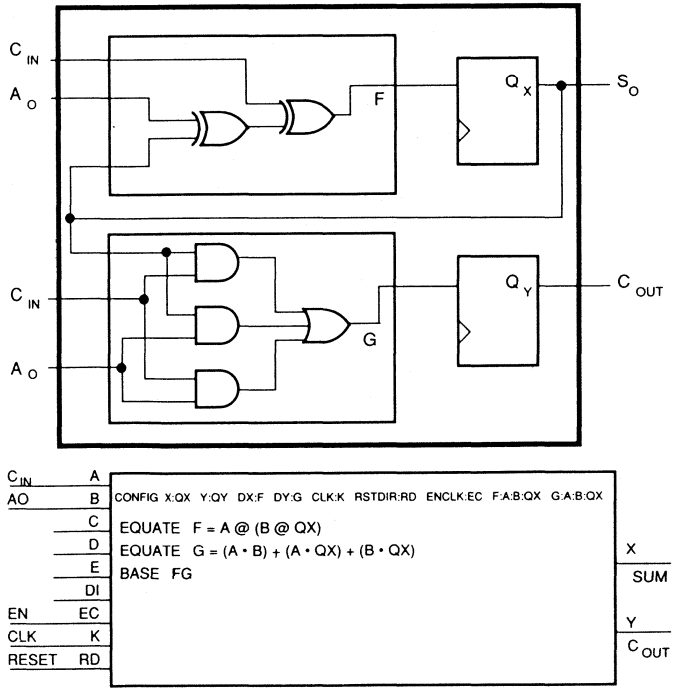
The XC3000 family can implement a very fast (>50MHz) accumulator with pipelined carry.

One CLB per bit stores the sum and the carry in its two flip-flops.

Each clock pulse updates the two flip-flops with the result of the addition of incoming operand plus stored sum.

There is, however, one drawback to this pipelined approach: An n-bit accumulator will need up to n-l additional clock pulses after the last accumulation in order to flush out the carry flip-flops.

PA



Programmable Sine Wave Generator

Sine wave frequency synthesizers are used in many applications, like telecom and navigation. A sine wave of programmable frequency can be generated by sequencing through a look-up table in ROM that drives a digital-to-analog converter (DAC).

The simplest and most flexible arrangement uses an accumulator to access the look-up table. (Remember, an accumulator is an adder/register structure that adds an input value to the register content each time it is clocked.) The desired frequency is presented as a constant (K) to the accumulator input. Changing K results in an instantaneous frequency change (as a result of the next clock edge) but no sudden phase change, no "clicks." This is mandatory in modems.

Here is one design example that fits into 30 CLBs, less than half of an XC3020: The objective is to generate any frequency that is an integer multiple of 1 Hz, the highest frequency being around 250 kHz. The sine wave look-up table has 64 entries for a 2π (360°) period, i.e. a resolution of 5° to 6°. It represents the amplitude as a 9-bit binary word (8 bits plus sign). These are reasonable parameters, but each of them could easily be modified by an order of magnitude without changing the design concept. The look-up table consists of a 64 x 8 ROM (really a 16 x 8 ROM plus XORs on the address inputs and data outputs) addressed by the 6 most significant outputs of the accumulator.

The ratio of max frequency to frequency resolution determines the size of the accumulator; in this case it is 250 kHz ÷ 1Hz = 250,000 or 18 bits. That would, however, give only one look-up per period at the top frequency; this design, therefore, adds four bits to the accumulator in order to guarantee sixteen look-ups even at 250 kHz. The accumulator clock rate is then determined by the frequency resolution (1 Hz) and the accumulator length (22 bits): If the accumulator increments by one for every clock period, it must step through the whole look-up table once per second. The clock frequency is, therefore, 2^{22} Hz = 4.194304 MHz.

The four most significant accumulator bits have no data inputs; they can, therefore, be implemented as a counter. The look-up table stores only the first quadrant (90°) of a sine wave, the other three quadrants are generated by reversing the address sequence (XORing the addresses) and/or reversing the sign of the output (XORing the outputs).

Better frequency resolution can be achieved by adding stages to the LSB end of the accumulator (1 CLB for each doubling of the resolution.) Same clock frequency.

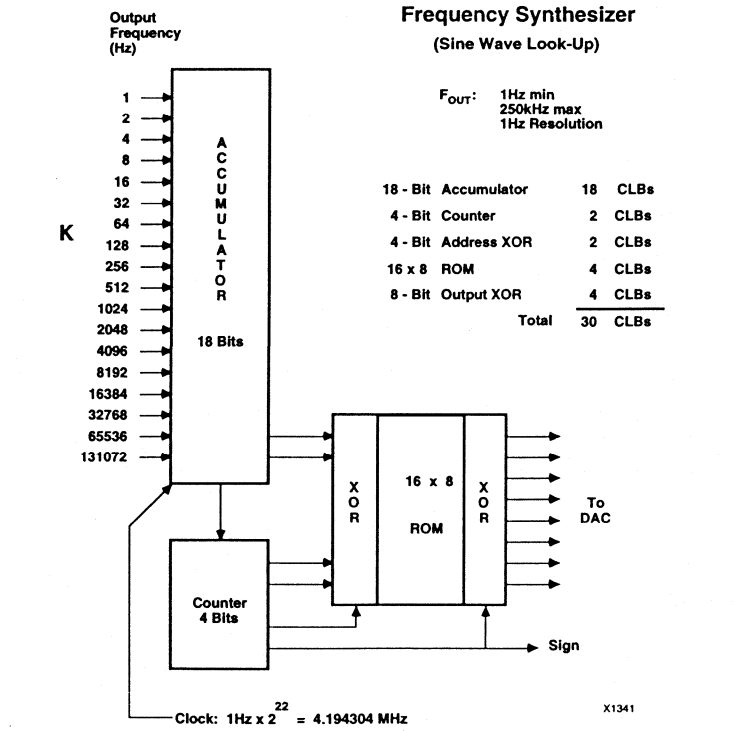
Higher max frequency can be achieved by adding to the MSB end of the accumulator and doubling the clock frequency for every additional bit.

The time granularity of the look-up table can be doubled to 32 entries per quadrant, increasing the table from 4 CLBs to 8.

The amplitude granularity of the look-up table can be changed in either direction by changing the number of look-up table planes.

Obviously, the look-up table can also store other wave shapes and can be reprogrammed dynamically.

These hints should allow any designer to custom tailor a similar frequency synthesizer. PA



No Can Do

Xilinx LCAs offer a wide range of design options and many system-oriented features. There are, however, some restrictions.

Here are the things you should not even try to do in the XC3000 family:

The on-chip input pull-up resistor cannot be used if the pin is configured as I/O, i.e., if the configuration allows the output to be activated. The resistor cannot be used to pull up a 3-stated output, use an external resistor instead.

Bidirectional buses are limited to the length of one Horizontal Longline. There is no way to interconnect bidirectional buses. There is no pass-transistor between the buses, and two back-to-back amplifiers would latch up.

IOB flip-flops and latches can be reset only by the global $\overline{\text{RESET}}$ package pin that resets every flip-flop and latch on the chip. Clock polarity is determined at the sources of the IOB clock line, not at each individual IOB.

IOB latches driven from the same clock line as a flip-flop have a surprising latch enable polarity: Active Low latch enable if the flip-flop clocks on the rising edge; active High latch enable if the flip-flop clocks on the falling edge. This enable polarity must be specified explicitly to avoid a "fatal DRC error".

The two flip-flops in a CLB cannot have separate clocks, clock enable or asynchronous reset inputs. The global clock distribution network cannot be used for anything else but driving CLB and IOB clock inputs. The alternate clock network, however, has limited access to the general-purpose interconnects.

PA

Volatility

Xilinx FPGAs use latches to store the data determining logic configuration and interconnects. Configuration information is written into these latches after power has been applied to the device, or whenever a re-configuration is initiated. Obviously, all configuration information is lost if power is interrupted. Some users have voiced concern about this. Here is a detailed explanation.

Configuration information remains valid provided Vcc stays >2.0 V. XC3000 and XC4000 devices, however, have an internal sensor that detects a Vcc drop below a critical value (~3 V). Even though the configuration is valid when that trip point is reached, the device goes into shut-down mode where it 3-states all outputs and clears the configuration memory, preparing it for a reconfiguration when Vcc returns to a more normal value.

There is no possibility of a Vcc dip causing the device to malfunction, i.e., to operate with erroneous configuration information.

- If Vcc stays above the trip point, the device functions normally, albeit at reduced speed, like any other CMOS device.
- If Vcc dips below the trip point, the device 3-states all outputs and waits for reconfiguration.

Some users feel uncomfortable with logic and interconnects defined by the content of latches. There is a concern about accidental or spontaneous changes. Xilinx designers have addressed these concerns. The Xilinx configuration storage latches are simple and rugged, far more rugged than the latches used in typical SRAMs.

Xilinx configuration latches consist of **cross-coupled inverters** with active pull-down n-channel and active pull-up p-channel transistors. The High and the Low level are thus both defined with active devices, each having an impedance of ~5 kΩ. Typical SRAMs use passive polysilicon pull-up resistors with an impedance of about 5,000 MΩ. A current of one nanoamp (!) would be sufficient to upset the typical SRAM cell, whereas it would take a million times more current to upset the Xilinx configuration latch.

This does not mean that SRAMs are unreliable, it just shows that the levels in Xilinx configuration latches are six orders of magnitude more resistant to upsets caused by external events, like cosmic rays or alpha particles. Xilinx has never heard about any occurrence of a spontaneous change in the configuration store in any of the 19 million LCA devices sold over the past eight years.

Xilinx production-tests the Vcc-dip tolerance of all XC3000 devices in the following way.

- After the device is configured, Vcc is reduced to 3.5 V, and then raised back to 5.0 V. Configuration data is then read back and compared against the original configuration bit stream. Any discrepancy results in rejection of the device.
- Subsequently, Vcc is reduced to 1.5 V and then raised to 5.0 V. The device must first go 3-state, then respond with a request for re-configuration.

Both these tests are performed at high temperature (>70°C for commercial parts, >125°C for military). Any part failing any of these tests is rejected as a functional failure.

PA

270-MHz Presettable Counter in XC3000

Prescaling is an established technique for high-speed counters. Using a derivative of this technique, LCA devices can implement a presettable counter at the full 270-MHz max toggle rate of the new XC3100-3. These counters can be up to 24-bits long.

In a prescaler counter, a small, very fast counter divides the clock rate. The divided clock is provided to a large, slower counter that is unable to settle at the fast clock rate. However, even when implemented synchronously, a conventional prescaler counter cannot be loaded; the technique depends upon the predictable binary sequence to ensure that the larger counter has adequate settling time.

If the prescaler counter is loaded with an arbitrary value, the binary sequence is broken, and the settling time of the larger counter is no longer guaranteed. To ensure an adequate settling time, either the clock frequency must be reduced significantly, or the values that can be loaded must be severely restricted.

To provide presettable prescaler counters, John Nichols of Fairchild Applications introduced a pulse-swallowing technique in 1970. It uses a dual-modulo prescaler that can divide the clock by 2^n or 2^n+1.

Twenty years later, Xilinx developed a variation of the pulse-swallowing technique for use in LCA devices. This technique, called state-skipping, uses a dual modulo prescaler that can divide by 2^n or 2^n-1.

In a state-skipping counter, the prescaler is not loaded. Instead, the least significant bits of the load value are used to initiate a correction counter that controls the modulus of the prescaler. Consequently, the larger counter, that contains the more significant bits, always has at least 2n-1 clock periods in which to settle, even after a load.

Typically, the minimum of 2^n-1 clock periods between the load and the first clock to the larger counter is longer than is required. To compensate, the prescaler operates with its shorter cycle until any extra delay has been nullified. This compensation is controlled automatically by the correction counter.

For example, in a counter using +7/+8 prescaler, the value loaded might require the first clock to the larger counter occur 5 clock periods after the load. In this case, the minimum 7-clock cycle period of the prescaler delays the first clock to the larger counter by two periods.

Excessive Idle Power in XC2000

Some users report a quiescent Icc consumption of more than 10 mA in the XC2000 family. This is usually the result of floating input pads, especially unbonded ones, and it can be fixed quite easily.

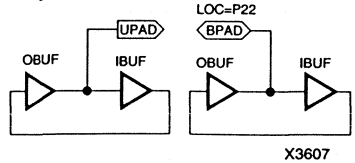
While the XC3000 and XC4000 devices have default pull-up resistors on all inputs, the XC2000 family lacks this option. Each **unused** pin or pad must, therefore, be forced to a valid logic level, either by an external connection or resistor, or by using its own output driver. The Makebits Tie option does **not** take care of this, it only ties **internal** inputs and interconnects to a defined logic level.

Users tend to overlook the **unbonded** pads on XC2064 PC44, XC2018 PC44 and XC2018 PC68. These devices have more internal pads than there are package pins available. Some pads are, therefore, left unbonded, but these unbonded inputs must also be forced to a valid logic level. Otherwise they might cause uncontrolled power consumption, and even uncontrolled oscillations. Unused outputs, bonded or not, can be tied from the schematic diagram. To do this, create dummy bidirectional pins using OBUFs, IBUFs, and BPADs or UPADs, as appropriate. Connect the OBUF output directly to IBUF input. This connection creates a tie circuit that is an input-less latch with no input. After configuration, these tie-circuits attain an unspecified, but well-defined logic level, and remain there, thus preventing the input from floating.

In existing designs, tie-circuits locked to unused pins can be added to the schematic, which is then recompiled using the previous LCA file as a guide. In new designs, an appropriate number of bonded-pin tie circuits included in the schematic will automatically be distributed among the unused bonded pins.

Tie-circuits cannot be locked to specific unbonded pins. However, if the correct number of unbonded tie circuits are included in the schematic, all unbonded pads will be tied. Tie circuits may also be added by editing the LCA design in XDE.

Unused bonded outputs tied in this way are active pins, and cannot, therefore, be used as PCB feedthroughs. However, unused pins required as feedthroughs need not be tied in the LCA device, since they do not float. PA



To nullify this extra delay, the prescaler continues dividing by 7 for a further two cycles, canceling one clock period of the extra delay each cycle. The third clock to the larger counter occurs 21 periods after the load, which is the same as in a conventional counter (5 + 8 + 8 = 21 clocks). Once the compensation is complete, the prescaler returns to dividing by 8.

Clearly, the counter will operate in a non-binary manner while the correction is being made. During this time, the counter skips a state each cycle of the prescaler, hence the name of the technique. The maximum time to complete the correction is 2^n-1 cycles of the prescaler. A further consequence of state-skipping is that some small division ratios cannot be used, because the correction cannot be completed within the period of the counter. In addition, the load must be synchronized with the prescaler cycle. This happens automatically if the counter is loaded when it reaches TC. This is common practice for timers and dividers, which are excellent application for state-skipping counters. With these exceptions, a state skipping counter may be loaded exactly like a conventional binary counter. There is no need to modify the load value required for any given divide ratio, as is necessary with a pulse-swallowing counter. One advantage of the state skipping technique that is peculiar to LCA implementation, is that a +3/+4 prescaler can be built in a single CLB. This is the key to the 270-MHz presettable counter, shown in the figure. The counter uses two state-skipping prescalers in cascade. Each is a 2-bit dual-modulo prescaler that divides by 3 or 4, and each has its own correction counter. Only the first prescaler is clocked by the high-speed clock. The maximum clock rate to the remainder of the counter is at least three times slower.

The first prescaler is implemented in a single CLB, and the counter design allows the control inputs several clock cycles to set up. Consequently, the high-speed clock is limited only by the toggle rate of the flip-flops in this CLB. In an XC3100-3 this is 270 MHz.

The remaining counters, including the first correction counter, are all clocked by Q_1. This synchronous operation permits the correction counters and Q_4 - Q_{23} to be loaded by Terminal Count in a conventional way.

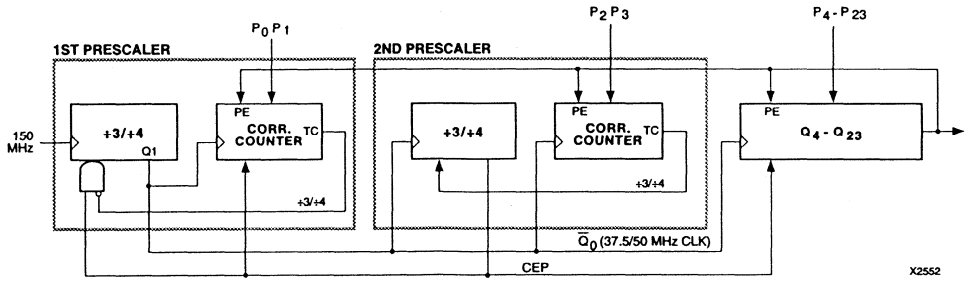
In each cycle of the second prescaler, only one of the three or four first-prescaler cycles can be a correction cycle. Consequently, the divide ratios of the composite prescaler are limited to 11,12,15 and 16, depending on which prescalers are correcting. This gives the Q_4 - Q_{23} counter at least 11 clock cycles in which to settle, and distribute the parallel enable signal.

Each time a prescaler correction cycle occurs, the corresponding correction counter is decremented. Correction cycles continue while the correction counters are non-zero. When zero is reached in either of the correction counters, the corresponding prescaler ceases correcting, and that correction counter remains at zero until it is reloaded.

Correction can take up to 45 clock periods to complete, and during this time some counter values will be skipped. However, the counter behaves in a conventional binary manner after less than 46 clock cycles. Some divide ratios below 30 cannot be used, since the correction time is greater than the counter period, but all divide ratios of 30 or greater are available.

State-skipping counters are the subject of an upcoming series of Applications Notes. Design files for the 24-bit 270-MHz Presettable Counter are available as XAPP021.

BN



270-MHz Counter

Simple RC Oscillator

This simple RC oscillator uses the XTAL2 or TCLK input, both of which guarantee a CMOS input threshold. The two counter-phase outputs are driven in such a way that the inverting output is derived from the non-inverting output. This prevents any possibility of spurious oscillations or erroneous operation.

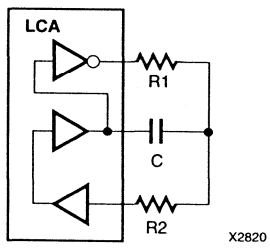
R1 is the main timing resistor. R2 allows the timing node to swing below ground and above Vcc.

$$f \approx 1/2RC \qquad R=R_1=R_2$$

The resistors can be any value between 500 Ω and 1 MΩ. At the low end, the output current becomes excessive; at the high end, the input leakage current affects the frequency, although this leakage current is really much lower than the 10-µA specification.

The capacitor can have any value between 50 pF and 1 µF. At the low end, the input capacitance affects the frequency; at the high end, the capacitor may become too big and too expensive. It is better to use an internal binary divider to extend the frequency to a lower value.

PA



Implementing ROM in XC3000 Devices

The RAM/ROM feature of the XC4000 CLB is well-known and supported by software. XC4000 CLBs, however, differ from other CLBs only in their ability to write data, thus providing a RAM. All LCA function generators can be used as ROMs, including those in the XC3000.

While table entry, such as provided by MEMGEN, is not supported in XC3000, two new Xilinx library symbols permit convenient entry of ROM data in the schematic. In addition to the expected address inputs and data outputs, the new symbols have 32 data inputs, one for each bit of ROM. Data is entered into the ROM by simply connecting these inputs to VCC or GND as desired.

Internally, the 16 x 2 ROM macro, Figure 1, is a dual 16:1 multiplexer. Connecting VCC and GND to the multiplexer inputs creates a function that is logically equivalent to the required ROM. A CLBMAP in the macro causes XACT to implement the multiplexer in a single CLB. The VCC and GND connections are absorbed into the function-generator look-up table, leaving only the address inputs and data outputs. Data inputs to the macro should only be connected to VCC or GND. Other signals will prevent the conversion to a single CLB.

The ROM access time is TILO, the combinatorial delay of the device. Viewlogic symbols are available for both 16 x 2 and 32 x 1 ROMs. They are named X3K_16X2 and X3K_32X1, and may be downloaded from the Xilinx Technical Bulletin Board Service (408-559-9327).

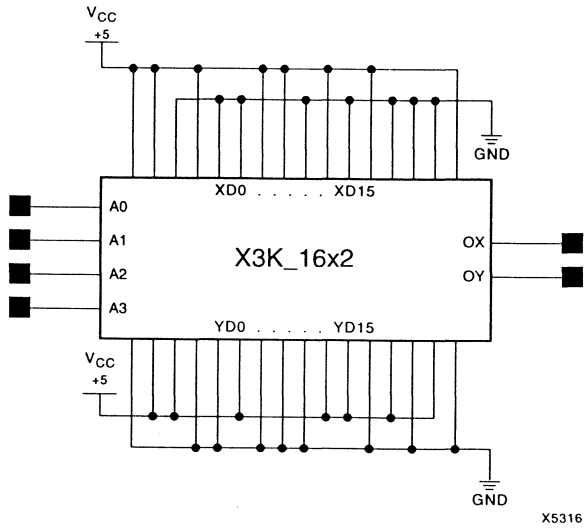


Figure 1. 16x2 ROM macro for the XC3000 series showing example connections of VCC and GND to the multiplexer inputs.

LCA Output Characteristics

Here are the first results of our output characteristics plotting.

Note that one device always represents a whole family, there is no difference between, e.g., XC3142 and XC3190 outputs.

Note that the XC4000 and XC7300 have n-channel-only outputs that do not drive any current above 3.5 V.

When pulling a Low output slightly below Ground, or a High output slightly above Vcc, the output impedance is the same as it is on the other side of Ground and Vcc, i.e., the plot shows a straight line going through Ground and Vcc. (The current direction changes, of course.)

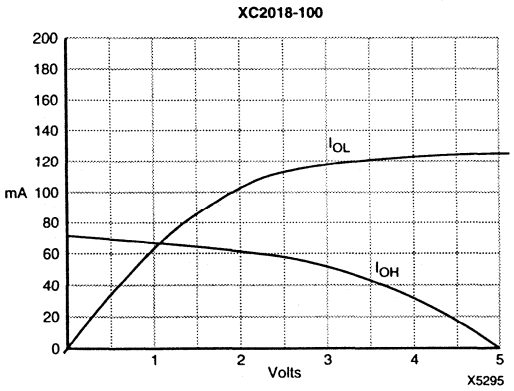
When the voltage exceeds 0.5 V below Ground or 0.5 V above Vcc, the protective diodes become conductive, and the current increases **dramatically**. That's why we should not specify a max voltage excursion, but rather a max current excursion into the forbidden territory below ground and above Vcc.

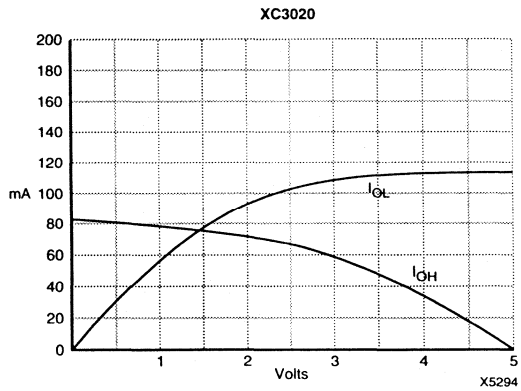
This is true for all devices. **Even XC4000 outputs have a strong clamp diode against Vcc.** Disregard previous statements to the contrary.

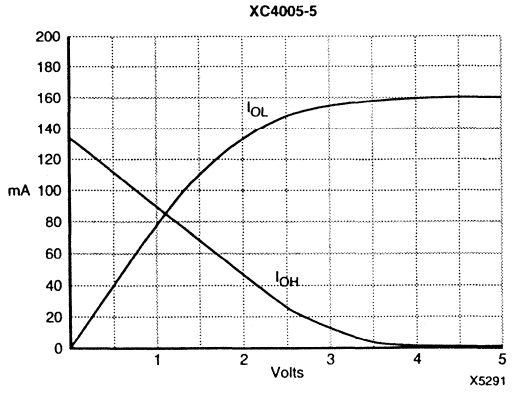
All measurements at 25°C and Vcc=5.00 V

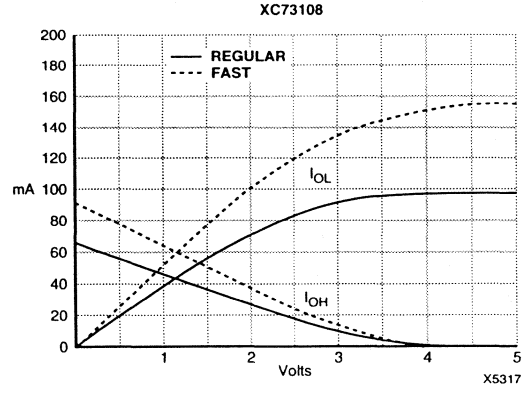
All parts are 1993 production type. PA

	Sink Current and Output Low Impedance			Source Current and Output High Impedance			
Device	**1 V**	**2 V**	**Impedance**	**4V**	**3V**	**2V**	**Impedance**
XC2018	70	>100 mA	14 Ω	-30	-52	-60 mA	35 Ω
XC3020	55	100 mA	20 Ω	-35	-60	-75 mA	30 Ω
XC4005	80	135 mA	13 Ω	0	-12	-50 mA	25 Ω
XC73108, reg.	40	70 mA	27 Ω	0	-10	-26 mA	40 Ω









Linear Feedback Shift Register Counters

Conventional binary counters use complex or wide fan-in logic to generate high end carry signals. A much simpler structure sacrifices the binary count sequence, but achieves very high speed with very simple logic, easily packing two bits into every CLB. Linear Feedback Register(LFSR)counters are also known as pseudo-random sequence generators.

An n-bit LFSR counter can have a maximum sequence length of 2^n-1. It goes through all possible code permutations except one, which is a lock-up state. A maximum length n-bit LFSR counter consists of an n-bit shift register with an XNOR in the feedback path from the last output Qn to the first input Dl. The XNOR makes the lock-up state the all-ones state; an XOR would make it the all-zeros state. For normal Xilinx applications, all-ones is preferred, since the flip-flops wake up in the all-zeros state.

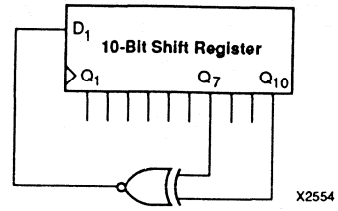
The table below describes the outputs that must drive the inputs of the XNOR. A mutli-input XNOR is also known as an even-parity circuit. Note that the connections described in this table are not necessarily unique. Due to the symmetry of the shift register operation and the XNOR function, other connections may also result in maximum length sequences.

Examples

- A 10-bit shift register counts modulo 1023, if the input Dl is driven by the XNOR of Q10 and the bit three positions to the left (Q7), i.e. a one is shifted into Dl when Q10 and Q7 have even parity, which means they are identical.
- An 8-bit shift register counts modulo 255 if the input Dl is driven by the XNOR of Q8, Q6, Q5, Q4, i.e., a one is shifted into Dl if these four outputs have even parity, (four zeros, or two ones, or four ones).

PA

n	XNOR Feedback from Outputs
3	3,2
4	4,3
5	5,3
6	6,5
7	7,6
8	8,6,5,4
9	9,5
10	10,7
11	11,9
12	12,6,4,1
13	13,4,3,1
14	14,5,3,1
15	15,14
16	16,15,13,4
17	17,14
18	18,11
19	19,6,2,1
20	20,17
21	21,19
22	22,21
23	23,18
24	24,23,22,17
25	25,22
26	26,6,2,1
27	27,5,2,1
28	28,25
29	29,27
30	30,6,4,1
31	31,28
32	32,22,2,1
33	33,20
34	34,27,2,1
35	35,33
36	36,25
37	37,5,4,3,2,1
38	38,6,5,1
39	39,35
40	40,5,4,3



Master & Slave Configure Together

All LCA users should know that daisy-chained devices automatically finish configuration together, and become active simultaneously. Each device counts all CCLK pulses, and each device has its own identical copy of the common length count value. When the number of CCLK pulses received equals this value, all devices in the daisy chain start up together. After a certain number of CCLK pulses, as determined by configuration options, all DONE pins go High, all RESETs are released, and all outputs go active simultaneously. This CCLK-driven synchronous start-up is automatically performed by the configuration control logic.

This information is not new, we only repeat it here because we got some phone calls that showed unnecessary concern on the part of the user.

PA

Legal Protection for Configuration Bit-Stream Programs

The bit-stream program loaded into the LCA may qualify as a "computer program" as defined in Section 101, Title 17 of the United States Code, and as such may be protectable under the copyright law. It may also be protectable as a trade secret if it is identified as such. We suggest that a user wishing to claim copyright and/or trade secret protection in the bit stream program consider taking the following steps.

1. Place an appropriate copyright notice on the LCA device or adjacent to it on the PC board to give notice to third parties of the copyright. For example, because of space limitations, this notice on the LCA device could read "©1992 XYZ Company" or, if on the PC board, could read "Bit Stream © 1992 XYZ Company".
2. File an application to register the copyright claim for the bit-stream program with the U.S. Copyright Office.
3. If practicable, given the size of the PC board, notice should also be given that the user is claiming that the bit-stream program is the user's trade secret. A statement could be added to the PC board such as: "Bit-stream proprietary to XYZ Company. Copying or other use of the bit-stream program except as expressly authorized by XYZ Company is prohibited."
4. To the extent that documentation, data books, or other literature accompanies the LCA device containing the bit-stream program, appropriate wording should be added to this literature providing third parties with notice of the user's claim of copyright and trade secret in the bit-stream program. For example, this notice could read: "Bit-Stream © 1992 XYZ Company. All rights reserved. The bit-stream program is proprietary to XYZ Company and copying or other use of the bit-stream program except as expressly authorized by XYZ Company is expressly prohibited."
5. To help prove unauthorized copying by a third party, additional non-functional code should be included at the end of the bit-stream program. Therefore, should a third party copy the bit-stream program without proper authorization, if the non-functional code is present in the copy, the copier cannot claim that the bit-stream program was independently developed.

These are only suggestions and Xilinx makes no representations or warranties with respect to the legal effect or consequences of the above suggestions. Each user is advised to consult legal counsel with respect to seeking protection of the user's bit-stream program and to determine the applicability of these suggestions to the user's products and circumstances.

If the user has any questions, contact the Xilinx legal department at 408-879-4984.

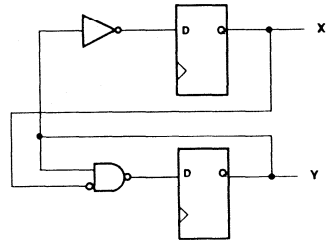
BH

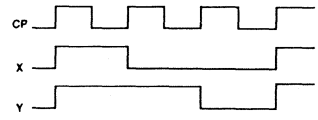
High Speed ÷3 Counter in One CLB

Some microprocessors require a 2/3 duty cycle clock, most conveniently and reliably generated by dividing a three times faster crystal oscillator frequency by three.

The design described below uses one XC3000 series CLB to generate a 1/3 High duty-cycle signal on the X output, and a 2/3 High duty cycle signal on the Y output. This is just one of many possible implementations. Max clock frequency is 100 MHz in a -125 device.

PA

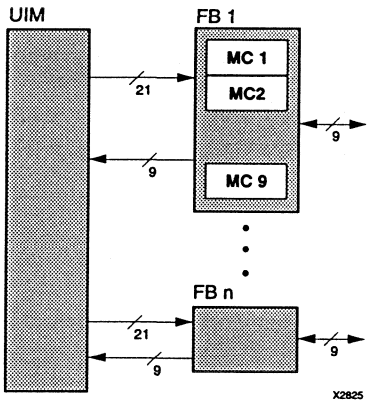




Anatomy of the EPLD Architecture

The XC7200 Architecture

The XC7200 devices provide multiple Function Blocks (FBs) interconnected by a central Universal Interconnect Matrix (UIM™). Each FB receives 21 signals from the UIM and produces nine output signals to pins and back into the UIM.

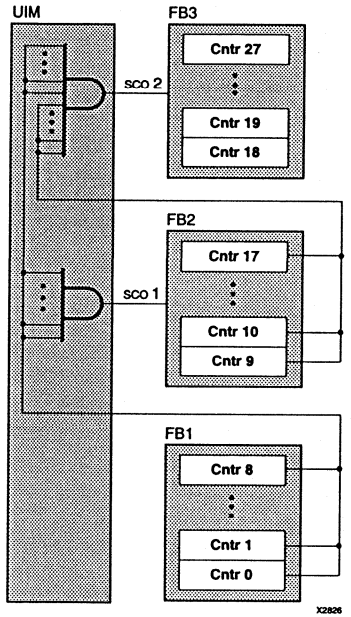


Simplified XC7200 Architecture (n=4,8,..)

Within each FB there are nine macrocells, each driven by product terms derived from the 21 UIM inputs. Each macrocell resembles a 21V9 PLD architecture. In addition, each macrocell includes an Arithmetic Logic Unit (ALU) that can generate and propagate arithmetic-carry signals between adjacent macrocells and Functional Blocks.

Universal Interconnect Matrix

Unlike other interconnect techniques, Xilinx EPLD's Universal Interconnect Matrix (UIM) provides 100% interconnectivity. Any output of any Function Block can be connected to any input or any number of inputs of any other Function Block using the UIM. The patented interconnect scheme of the UIM provides a fast uniform delay through any of its paths. This interconnect is independent of fan-in or fan-out loading. Because each FB has identical timing characteristics and the UIM has a constant delay, logic mapped into the device has predictable performance, independent of placement and routing. The UIM can also act as one or more AND gates, e.g., to form terminal count signals within the interconnect. The following diagram illustrates how 27 macrocells can be configured to implement a 27-bit synchronous counter.



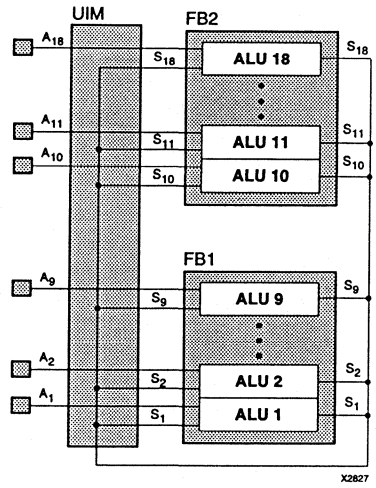
27-bit Synchronous Counter Implemented Using UIM AND Gates

Consistent timing performance for all on-chip signals greatly simplifies the design process. In addition to interconnection, the UIM is also used for the following functions:

- Emulating 3-state buses
- Enable/disable signal gates
- Logic decoders
- DeMorgan OR gates

Arithmetic Logic Unit (ALU)

Unlike other programmable logic arrays, the XC7200 architecture includes dedicated arithmetic logic units and fast carry lines running directly between adjacent macrocells and Function Blocks. This additional ALU enables the XC7200 architecture to support fast adders, subtractors, and magnitude comparators of any length up to 72 bits. The following diagram illustrates how 18 macrocells (2 Function Blocks) can be programmed to implement an 18-bit accumulator.



18-bit Accumulator Implemented using ALU Chain

The above architectural features introduce innovative systems-oriented enhancements to the classical features of the PAL-like CPLD architectures. This favors the implementation of fast state machines, large synchronous counters and fast arithmetic, as well as multi-level general-purpose logic. DR

Dynamic Power Consumption

It is impossible to give a max value for LCA power consumption, because it is totally dynamic. The power consumption of any node is proportional to its capacitance multiplied by the frequency at which it is charged and discharged between +5 V and ground. To determine total power, you must know the capacitance of each node and clock line inside the chip, and the frequency with which it is moving up and down; and, you must know the external capacitive load and its frequency.

A worst-case maximum number would be very high, and therefore meaningless, because nobody designs a system where every node moves at 60 MHz, for example.

Estimating power consumption usually has one of two goals: Thermal reliability evaluation, or power-supply sizing.

Thermal calculations can often be substituted by rough estimates, since CMOS power is so low. We give Θ_{JA} values for each package, describing the thermal impedance, i.e. the temperature rise in °C per Watt of power dissipation. Assuming a very conservative max junction temperature of 145°C, a max ambient temperature of 60°C, and a Θ_{JA} of 30°C/W gives a max allowable power dissipation of 2.8 W. Very few LCA designs consume that amount of power, most use a few hundred milliwatts, which results in a junction temperature only a few degrees above ambient.

LCA devices are usually not the dominating power consumers in a system, and do not have a big impact on the power supply design. There are, of course, exceptions to these general rules, and the designer should then use the data on this page to estimate power consumption more accurately. PA

Here are the results of recent measurements of the dynamic power consumption in various Xilinx devices.

The Applications section of our Data Book describes the same parameters, but those values were based on 1988 measurements of devices with larger geometries.

XC2018 at 5.0 V

One CLB driving 3 local interconnects	0.22 mW / MHz
One device output with a 50 pF load	2.0 mW / MHz
One Global clock buffer & line	3.2 mW / MHz

XC2018 at 3.3 V

One CLB driving 3 local interconnects	0.1 mW = 0.03 mA/MHz
One device output with a 50 pF load	0.8 mW = 0.35 mA/MHz
One Global clock buffer & line	1.0 mW = 0.3 mA/MHz

XC3020

One CLB driving 3 local interconnects	0.25 mW / MHz
One device output with a 50 pF load	1.25 mW / MHz
One Global clock buffer & line	2.0 mW / MHz
One Longline without driver	0.1 mW / MHz

XC3020L at 3.3 V

One CLB driving 3 local interconnects	0.1 mW/MHz = 0.03 mA/MHz
One device output with a 50 pF load	0.5 mW/MHz = 0.15 mA/MHz
One Global clock buffer & line	0.8 mW/MHz = 0.25 mA/MHz
One split Longline without driver	0.04 mW/MHz = 0.01 mA/MHz

XC3090

One CLB driving 3 local interconnects	0.25 mW / MHz
One device output with a 50 pF load	1.25 mW / MHz
One Global clock buffer & line	3.5 mW / MHz
One split Longline without driver	0.15 mW / MHz

XC4003

One CLB driving 3 local interconnects	0.30 mW / MHz
One device output with a 50 pF load	1.2 mW / MHz
One Global clock buffer & line	1.9 mW / MHz
One split Longline without driver	0.12 mW / MHz

XC4005

One CLB driving 3 local interconnects	0.30 mW / MHz
One device output with a 50 pF load	1.2 mW / MHz
One Global clock buffer & line	3.2 mW / MHz
One split Longline without driver	0.17 mW / MHz

XC4010

One CLB driving 3 local interconnects	0.30 mW / MHz
One device output with a 50 pF load	1.2 mW / MHz
One Global clock buffer & line	5.1 mW / MHz
One split Longline without driver	0.24 mW / MHz

PC-Board Design Hints for LCA Users

Twenty years ago, CMOS was hailed as the friendliest form of logic: no input current, full rail-to-rail logic swing, high noise immunity, soft edges, low power consumption, tolerance for large Vcc variations, etc.

Things have changed. CMOS devices have lost some of their user-friendliness as they have become faster and faster, partly in a deliberate quest for speed, partly as an unavoidable result of the smaller device geometries that are required to lower manufacturing cost. The output edge rate is now faster than for TTL, and PC-board interconnect lines between modern CMOS devices can no longer be treated a short circuits or lumped capacitances. The CMOS designer must now cope with the same transmission-line effects that concerned the previous generation of designers using Schottky-TTL or ECL devices.

Here are some basic rules.

Rule 1: Any PC-board trace is really a transmission line with distributed capacitance and inductance. The series resistance is usually unimportant. The table at left lists typical values for the capacitance and inductance of a PC-board trace with a ground-plane below it.

Any voltage change on such a transmission line causes a corresponding current change. The voltage-to-current ratio is called the characteristic impedance, Z_0 . It is determined by the line thickness and width, by the distance to the ground plane, and by the dielectric constant ε of the PC- board material. Z_0 is independent of line length. The table shows typical values for popular constellations.

Rule 2: Signals travel along a transmission line at roughly half the speed of light, or 6" (15 cm) per nanosecond. More precisely, the true propagation speed is the free-air speed of light divided by the square root of the effective dielectric constant, ε. The speed of light is very close to 12" (30 cm) per nanosecond, and ε for typical epoxy material is 4.7. Since some of the electric field passes through air, the effective ε is closer to 4, which leads to the rule of thumb,"half the speed of light".

Rule 3: Whenever the one-way propagation time along a wire or PC trace is longer than **half the rise or fall time of the driving signal**, this wire or trace must be considered a transmission line, not a lumped capacitive load.

If the rise- or fall-time is 1.5 ns, any PC-board trace longer than 4.5 inches (11cm) must be analyzed for transmission-line effects.

If the rise or fall time is 5 ns, only PC-board traces longer than 15" (38 cm) need to be analyzed for transmission-line effects.

When a fast rising edge is being driven onto a long transmission line, the driver sees the characteristic impedance Z_0 (50 to 150 Ω), and generates a voltage step that is determined by the ratio of output impedance, Ri, to Z_0. Typically, an LCA device with an output impedance of 60 Ω drives a 3.5-V step onto a 100-Ω line.

This step propagates to the end of the line at a speed of 6" (15 cm) per nanosecond. If the far end is left open or has a light capacitive load, e.g., the input to a CMOS device, a reflected wave is superimposed on the incoming wave, since only an equal-amplitude reflected wave satisfies the zero-current requirement at the end of the line. This reflected wave travels back to the signal source, arriving there with almost double the original amplitude, usually well above Vcc. If the output impedance of the driver differs from Z_0, the incoming wave is again reflected, travels to the far end, where it is reflected again, etc. This series of reflections with decreasing amplitude is commonly called "ringing". Theoretically, these are rectangular steps of alternating and decreasing amplitude, but high-frequency imperfections often give it the appearance of a decaying sine wave.

At best, such reflections result in poor signal quality and loss of noise immunity. At worst, they reduce system performance and cause functional failures due to double clocking.

	0.062" Board		0.031"Board	
Z_0 (Ω)	Width (mils)	C (pF/ft)	Width (mils)	C (pF/ft)
50	103	35	47	31
60	77	29	35	27
70	57	25	26	23
80	42	22	19	20
90	31	20	14	18
100	23	18	10	16

Imperial units

	1.6 mm Board		0.8 mm Board	
Z_0 (Ω)	Width (mm)	C (pF/cm)	Width (mm)	C (pF/cm)
50	2.6	1.15	1.2	1.02
60	2.0	0.95	0.9	0.88
70	1.4	0.82	0.65	0.75
80	1.1	0.72	0.5	0.66
90	0.8	0.66	0.35	0.59
100	0.6	0.6	0.25	0.52

Metric units

Microstrip-Line Impedance and Capacitance per Unit Length

Coping with Transmission Line Effects

Parallel Termination, Figure 2.

A transmission line of arbitrary length can be terminated at the far end by a resistor to ground or Vcc. If this resistor equals the characteristic impedance Z_0, the driver always sees the transmission line like a lumped resistive load. Any signal driven onto the line travels to the far end and is dissipated in the resistor. There is no reflection, no ringing or overshoot. Unfortunately, this type of termination is usually impractical, because it puts undue current and power requirements on the driver. It requires 100 mA to drive a 5 V signal onto a 50 Ω line. Only ECL circuits or special buffer circuits can drive terminated transmission lines conveniently. There are two popular methods to alleviate the problem.

- Connecting the terminating resistor through a fairly large capacitor to ground instead of directly to ground or Vcc, reduces static power consumption, but introduces a time constant that must be tailored to the system speed.
- Terminating the line with two resistors, one to ground and one to Vcc, reduces the peak current-requirement. 300 Ω to Vcc and 150 Ω to ground is the Thevenin equivalent of a 100 Ω termination to 1.6 V.

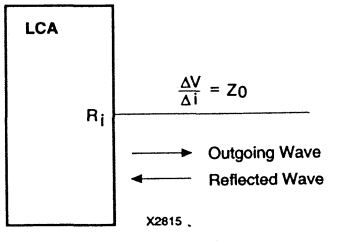


Fig. 1. Transmission Line

Series Termination, Figure 3.

In some cases, series termination at the source can offer the benefits of termination without the drawbacks mentioned above. When an additional series-resistor between the driver and the line increases the effective drive impedance to the same value as Z_0, the transmission line receives a starting step of half amplitude. Adding an external 40-Ω resistor to the 60-Ω LCA output impedance matches the 100-Ω transmission line, and drives it with a 2.5-V step. This step travels to the far end, where it is reflected and thus doubled in amplitude, as described above. It then arrives back at the driven end of the line with full amplitude (5 V), and is not reflected, since it sees a terminating resistor that is equal to Z_0.

This seemingly ideal solution has one big drawback: A half-amplitude voltage step travels along the trace and back. Everywhere along the line, except at the far end, this half-amplitude signal can cause trouble, especially in the vicinity of the driver. Series termination is, therefore, recommended only for signals that go from a **single source to a single destination**. Taps on a series-terminated have half-amplitude (2.5 V) levels for fairly long times, which means poor noise immunity and potential malfunction.

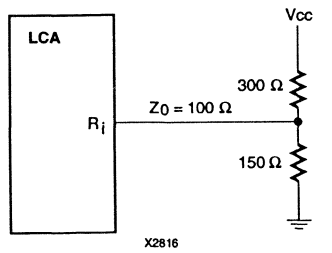


Fig. 2. Parallel Termination

Practical Rules

- Use slew-rate limited outputs wherever possible. Their longer rise and fall times eliminate transmission line effects for all short interconnects.
- Keep critical interconnects as short as possible. It may be better to duplicate some logic in the LCA device and drive from different sides of the device, if that shortens the PC-board traces.
- Use multi-layer PC boards with ground and Vcc planes whenever possible. Always connect all Vcc and ground pins, and be generous with Vcc decoupling capacitors, 0.1 µF per Vcc pin.
- Use series termination for lines that drive a single or lumped destination, but never put taps on a series-terminated line.
- In synchronous systems, the synchronous data and control lines can tolerate poor signal quality **after the clock edge**, but all asynchronous inputs, and especially all clock inputs need good signal quality **all the time.**
- Pay attention to clock distribution on the PC board. Low-skew drivers are now available, e.g., the NSC CGS74C2525.
- CMOS-level input threshold offers the best noise immunity. (Not available on XC4000).
- Remember that a low clock frequency does not make the circuits slow. When the system clock rate is very low, the flip-flops inside the LCA device can still react to 2-ns clock spikes.

PA

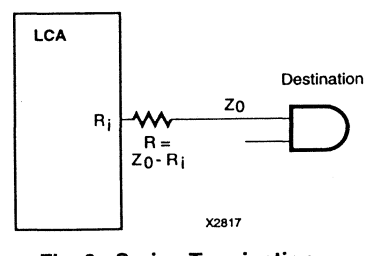


Fig. 3. Series Termination

Crystal-Oscillator Considerations

There are two reasons why many designers feel uncomfortable using the on-chip LCA crystal oscillator circuit.

- This is analog territory, unfamiliar to many digital designers. Words like reactance transconductance, gain, dB, phase response, jωL and s-plane evoke memories of long-forgotten early college classes.
- IC documentation is usually skimpy on the issue of specifying crystals and designing reliable oscillator circuits.

Here is additional information.

Let's start with some fundamental facts. There is nothing Xilinx-specific about the oscillator circuit. It's a wide-band inverting amplifier, as used in all popular microcontrollers, like the 8051. When a crystal and some passive components close the feedback path, as shown in our Data Book, this circuit becomes a reliable and stable clock source.

The path from XTAL2 to XTAL1 inside the LCA device (strangely enough, XTAL2 is the input, XTAL1 is the output) is a single-stage inverting amplifier, which means it has a low-frequency phase response of 180°, increasing by 45° at the 3-dB frequency.

Input impedance is 10-15 pF, input threshold is CMOS, but dc bias must be supplied externally through a megohm resistor from XTAL1 to XTAL2.

Low-frequency gain is about 20, rolling off 3dB at 125 MHz.

Output impedance is between 50 and 100 Ω and the capacitance on the output pin is 10 to 15 pF.

Pulse response is a delay of about 1.5 ns and a rise/fall time of about 1.5 ns.

For stable oscillation,

- the loop gain must be exactly one, i.e., the internal gain must be matched by external attenuation, and
- the phase shift around the loop must be 360° or an integer multiple thereof. The external network must, therefore, provide 180° of phase shift.

A crystal is a piezoelectric mechanical resonator that can be modeled by a very high-Q series LC circuit with a small resistor representing the energy loss. In parallel with this series-resonant circuit is unavoidable parasitic capacitance inside and outside the crystal package, and usually also discrete capacitors on the board.

The impedance as a function of frequency of this whole array starts as a small capacitor at low frequencies (Figure 1). As the frequency increases, this capacitive reactance decreases rapidly, until it reaches zero at the series resonant frequency.

At slightly higher frequencies, the reactance is inductive, starting with a zero at series resonance, and increasing very rapidly with frequency. It reaches infinity when the effective inductive impedance of the series LC circuit equals the reactance of the parallel capacitor. **The parallel resonance frequency is a fraction of a percent above the series-resonance frequency.**

Over this very narrow frequency range between series and parallel resonance, the crystal impedance is inductive and changes all the way from zero to infinity. The energy loss represented by the series resistor prevents the impedance from actually reaching zero and infinity, but it comes very close.

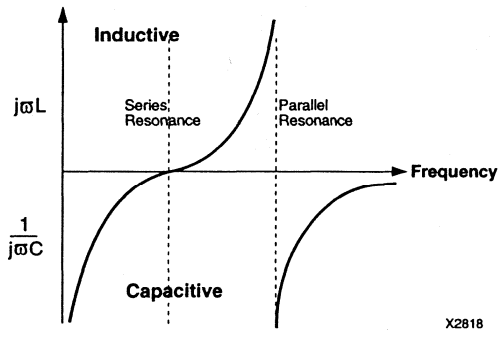


Figure 1. Reactance as a Function of Frequency

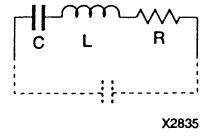


Figure 2. Equivalent Circuit

Microprocessor- and FPGA-based crystal oscillators all operate in this narrow frequency band, where the crystal impedance can be any inductive value. The circuit oscillates at a frequency where the attenuation in the external circuit equals the gain in the LCA device, and where the total phase shift, internal plus external, equals 360°.

Figure 3 explains the function. At the frequency of oscillation, the series-resonant circuit is effectively an inductor, and the two capacitors act as a capacitive voltage divider, with the center-point grounded. This puts a virtual ground somewhere along the inductor and causes the non-driven end of the crystal to be 180° out of phase with the driven end, which is the external phase shift required for oscillation. This circuit is commonly known as a Pierce oscillator.

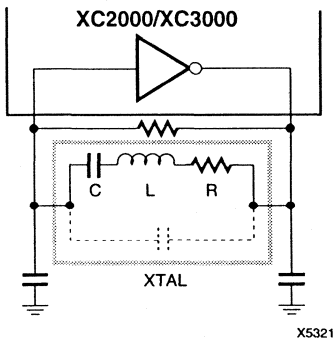


Figure 3. Pierce Oscillator

Practical Considerations

- The series resonance resistor is a critical parameter. To assure reliable operation with worst-case crystals, the user should experiment with a discrete series resistor roughly equal to the max internal resistance specified by the crystal vendor. If the circuit tolerates this additional loss, it should operate reliably with a worst-case crystal without the additional resistor.
- The two capacitors affect the frequency of oscillation and the start-up conditions. The series connection of the two capacitors is the effective capacitive load seen by the crystal, usually specified by the crystal vendor.
- The two capacitors also determine the minimum gain required for oscillation. If the capacitors are too small, more gain is needed, and the oscillator may be unstable. If the capacitors are too large, oscillation is stable but the required gain may again be higher. There is an optimum capacitor value, where oscillation is stable, and the required gain is at a minimum. For most crystals, this capacitive load is around 20 pF, i.e., each of the two capacitors should be around 40 pF.
- Crystal dissipation is usually around 1 mW, and thus of no concern. Beware of crystals with "drive-level dependence" of the series resistor. They may not start up. Proper drive level can be checked by varying Vcc. The frequency should increase slightly with an increase in Vcc. A decreasing frequency or unstable amplitude indicate an over-driven crystal. Excessive swing at the XTAL2 input results in clipping near Vcc and ground. An additional 1 to 2 kΩ series resistor at the XTAL1 output usually cures that distortion problem. It increases the amplifier output impedance and assures additional phase margin, but results in slower start-up.
- Be especially careful when designing an oscillator that must operate near the specified max frequency. The circuit needs excess gain at small signal amplitudes to supply enough energy into the crystal for rapid start-up. High-frequency gain may be marginal, and start-up may be impaired.
- Keep the whole oscillator circuit physically as compact as possible, and provide a single ground connection. Grounding the crystal can is not mandatory but may improve stability.

Series Resonant or Parallel Resonant?

Crystal manufacturers label some crystals as series-resonant, others as parallel-resonant, but there really is no difference between these two types of crystals, they all operate in the same way. Every crystal has a series resonance, where the impedance of the crystal is extremely low, much lower than at any other frequency. At a slightly higher frequency, the crystal is inductive and in parallel resonance with the unavoidable stray capacitance or the deliberate capacitance between its pins .

The only difference between the two types of crystal is the manufacturer's choice of specifying either of the two frequencies. If series resonance is specified, the actual frequency of oscillation is a little higher than the specified value. If parallel resonance is specified, the frequency of oscillation is a little lower. In most cases, these small deviations are irrelevant.

Sources

Fick: "Schwingquarz und Mikroprozessor". *Elektronik*, Feb. 1987

Horowitz & Hill: *The Art of Electronics*, Cambridge University Press, 1989

Motorola High-Speed CMOS Logic Data Book., 1983.

PA

Two's Complementer Packs 2 Bits per CLB

The best known algorithm for twos complementing a number is to invert all the bits and then add one to the result. Using this algorithm, only one data bit can be generated by each XC3000 CLB, since the increment operation requires an additional carry output for each bit. However, an alternate empirical algorithm exists that does not have this limitation, and generates two bits per CLB.

This alternate algorithm permits the two's complement of a number to be determined by inspection. As shown in the example, the number is scanned, one bit at a time, from the least significant end, until the first "one" is encountered. The first "one" and any less significant zeros are passed to the output unchanged. All more significant bits are inverted.

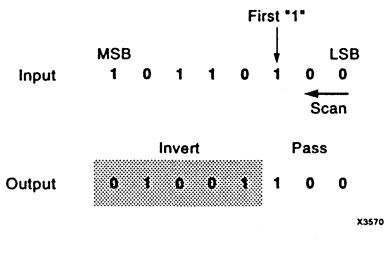
This algorithm may be rewritten in an iterative form: a bit is inverted only if its less significant neighbor is inverted, or is a one. Trailing zeros and the first "one" are not inverted because their less significant neighbors are neither inverted nor "ones". The bit after the first "one" is inverted because its neighbor is a "one", and the remaining bits are inverted because their neighbors were .

This may be implemented as shown in Figure 1. The inputs and outputs of the less significant neighbors are inspected to determine their value and whether they were inverted. An XOR is then used to invert the data according to the rule described above.

The least significant bit always remains unchanged when two's complementing a number. Consequently, no logic is required by the LSB and no less significant neighbor is required.

Figure 2 shows a modification of the 2-bit CLB that only incurs one delay per bit-pair. This doubles the performance of the original design, without increasing the number of CLBs required or the routing complexity.

A further modification, not shown, permits the delay for an 8-bit complementer to be reduced to two CLBs. However, one additional CLB is required. In this design, complementers larger than eight bits use two additional CLBs per three bits, and the delay increases by one CLB per three bits.



Two's Complement Example

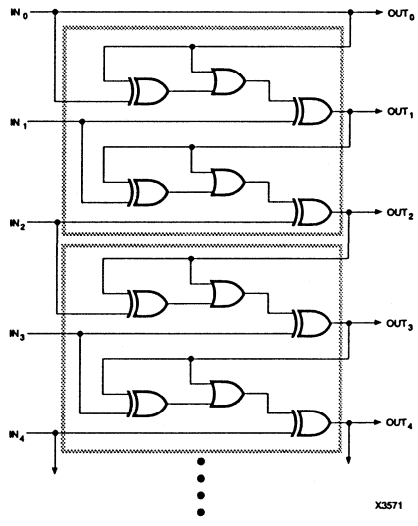


Figure 1. Simple Two's Complementer

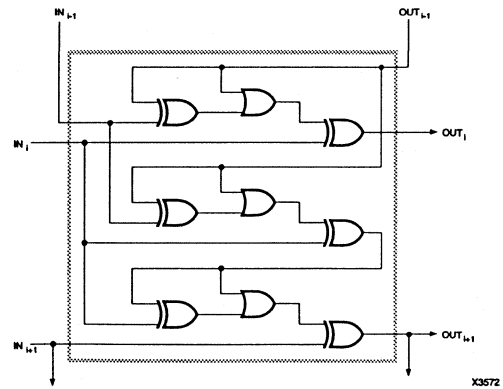


Figure 2. Faster Two's Complementer

Index, Sales Offices

Index

Sales Offices

HEADQUARTERS

XILINX, Inc.
2100 Logic Drive
San Jose, CA 95124
(408) 559-7778
TWX: 510-600-8750
FAX: 408-559-7114

XILINX SALES OFFICES

NORTH AMERICA

XILINX, Inc.
3235 Kifer Road
Suite 320
Santa Clara, CA 95051
(408) 245-1361
FAX: 408-245-0517

XILINX, Inc.
15615 Alton Parkway
Suite 280
Irvine, CA 92718
(714) 727-0780
FAX: 714-727-3128

XILINX, Inc.
61 Spit Brook Rd.
Suite 403
Nashua, NH 03060
(603) 891-1096
FAX: 603-891-0890

XILINX, Inc.
65 Valley Stream Parkway
Suite 140
Malvern, PA 19355
(610) 296-8302
FAX: 610-296-8378

XILINX, Inc.
939 North Plum Grove Road
Suite H
Schaumburg, IL 60173
(708) 605-1972
FAX: 708-605-1976

XILINX, Inc.
5952 Six Forks Road
Raleigh, NC 27609
(919) 846-3922
FAX: 919-846-8316

XILINX, Inc.
4141 Blue Lake Circle
Suite 217
Dallas, TX 75244
(214) 960-1043
FAX: 214-960-0927

EUROPE

XILINX, Ltd.
Suite 1B, Cobb House
Oyster Lane
Byfleet
Surrey KT14 7DU
United Kingdom
Tel: (44) 932-349401
FAX: (44) 932-349499

XILINX Sarl
Z.I. de la Bonde-Bât.B
1 bis, rue Marcel Paul
91742 Massy Cedex
France
Tel: (33)1-60-13-34-34
FAX: (33)1-60-13-04-17

XILINX GmbH
Dorfstr. 1
85609 Aschheim
Germany
Tel: (49) 89-904-5024
FAX: (49) 89-904-4748

JAPAN

XILINX K. K.
Daini-Nagaoka Bldg. 2F
2-8-5, Hatchobori, Chuo-ku
Tokyo 104
Japan
Tel: (81) 3-3297-9191
FAX: (81)3- 3297-9189

ASIA PACIFIC

XILINX Asia Pacific
Unit 2308-2319, Tower 1
Metroplaza
Hing Fong Road
Kwai Fong, N.T.
Hong Kong
Tel: 852-410-2740
FAX: 852-418-1600

U.S. SALES REPRESENTATIVES

ALABAMA

Novus Group, Inc. (Corporate)
2905 Westcorp Blvd.Suite 120
Huntsville, AL 35805
(205) 534-0044
FAX: 205-534-0186

ARIZONA

Quatra Associates
4645 S. Lakeshore Dr.
Suite 1
Tempe, AZ 85282
(602) 820-7050
FAX: 602-820-7054

ARKANSAS

Bonser-Philhower Sales
689 W. Renner Road
Suite 101
Richardson, TX 75080
(214) 234-8438
FAX: 214-437-0897

CALIFORNIA

SC Cubed
31225 La Baya Dr. #111
Westlake Village, CA 91362
(818) 865-6222
FAX: 818-865-6223

SC Cubed
17862 17th St. Suite 207
Tustin, CA 92680
(714) 731-9206
FAX: 714-731-7801

Quest-Rep Inc.
6494 Weathers Pl, Suite 200
San Diego, CA 92121
(619) 622-5040
FAX: 619-622-5047

Norcomp
1267 Oakmead Pkwy
Sunnyvale, CA 94086
(408) 733-7707
FAX: 408-774-1947

Norcomp
8880 Wagon Way
Granite Bay, CA 95746
(916) 791-7776
FAX: 916-791-2223

COLORADO

Luscombe Engineering, Inc.
1500 Kansas Ave. Suite 1B
Longmont, CO 80501
(303) 772-3342
FAX: 303-772-8783

CONNECTICUT

John E. Boeing, Co., Inc.
101 Harvest Park, Bldg. 1A
No. Plains Industrial Road
Wallingford, CT, 06492
(203) 265-1318
FAX: 203-265-0235

DELAWARE

Delta Technical Sales, Inc.
122 N. York Rd., Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

FLORIDA

Semtronic Assoc., Inc.
657 Maitland Avenue
Altamonte Springs, FL 32701
(407) 831-8233
FAX: 407-831-2844

Semtronic Assoc., Inc.
3471 N. W. 55th Street
Ft. Lauderdale, FL 33309
(305) 731-2484
FAX: 305-731-1019

Semtronic Assoc., Inc.
1467 South Missouri Avenue
Clearwater, FL 34616
(813) 461-4675
FAX: 813-442-2234

GEORGIA

Novus Group, Inc.
6115-A Oakbrook Pkwy
Norcross, GA 30093
(404) 263-0320
FAX: 404-263-8946

IDAHO (Southwest)

Thorson Company Northwest
12340 NE 8th St., Suite 201
Bellevue, WA 98005
(206) 455-9180
FAX: 206-455-9185

Luscombe Engineering, Inc.
670 East 3900 South #103
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

ILLINOIS

Beta Technology Sales, Inc.
1009 Hawthorne Drive
Itasca, IL 60143
(708) 250-9586
FAX: 708-250-9592

Advanced Technical Sales
13755 St. Charles Rock Rd.
Bridgeton, MO 63044
(314) 291-5003
FAX: 314-291-7958

INDIANA

Gen II Marketing,Inc.
31 E. Main St.
Carmel, IN 46032
(317) 848-3083
FAX: 317-848-1264

Gen II Marketing, Inc.
1415 Magnavox Way
Sutie 130
Ft. Wayne, IN 46804
(219) 436-4485
FAX: 219-436-1977

IOWA

Advanced Technical Sales
375 Collins Road N.E.
Cedar Rapids, IA 52402
(319) 393-8280
FAX: 319-393-7258

KANSAS

Advanced Technical Sales
610 N. Mur-Len, Suite B
Olathe, KS 66062
(913) 782-8702
FAX: 913-782-8641

KENTUCKY

Gen II Marketing, Inc.
4012 DuPont Circle
Suite 414
Louisville, KY 40207
(502) 894-9903
FAX: 502-893-2435

LOUISIANA (Northern)

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
FAX: 214-437-0897

LOUISIANA (Southern)

Bonser-Philhower Sales
10700 Richmond, Suite 150
Houston, TX 77042
(713) 782-4144
FAX: 713-789-3072

MAINE

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

MARYLAND

Micro Comp, Inc.
1421 S. Caton Avenue
Baltimore, MD 21227-1082
(410) 644-5700
FAX: 410-644-5707

Sales Offices

MASSACHUSETTS

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

MICHIGAN

Miltimore Sales Inc.
22765 Heslip Drive
Novi, MI 48375
(810) 349-0260
FAX: 810-349-0756

Miltimore Sales Inc.
3680 44th St., Suite 100-J
Kentwood, MI 49512
(616)-554-9292
FAX: 616-554-9210MINNESOTA
Com-Tek
6513 City West Parkway
Eden Prairie, MN 55344
(612) 941-7181
FAX: 612-941-4322

MISSISSIPPI

Novus Group, Inc.(Corporate)
2905 Westcorp Blvd. Suite 120
Huntsville, AL 35805
(205) 534-0044
FAX: 205-534-0186

MISSOURI

Advanced Technical Sales
601 N. Mur-Len, Suite 8
Olathe, KS 66062
(913) 782-8702
FAX: 913-782-8641

Advanced Technical Sales
13755 St. Charles Rock Rd.
Bridgeton, MO 63044
(314) 291-5003
FAX: 314-291-7958

MONTANA

Luscombe Engineering, Inc.
670 East 3900 South #103
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

NEBRASKA

Advanced Technical Sales
375 Collins Road N.E.
Cedar Rapids, IA 52402
(319) 393-8280
FAX: 319-393-7258

NEVADA

Norcomp
(Excluding Las Vegas)
3350 Scott Blvd., Suite 24
Santa Clara, CA 95054
(408) 727-7707
FAX: 408-986-1947

Quatra Associates
(Las Vegas)
4645 S. Lakeshore Dr., Suite 1
Tempe, AZ 85282
(602) 820-7050
FAX: 602-820-7054

NEW HAMPSHIRE

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

NEW JERSEY (Northern)

Parallax
734 Walt Whitman Road
Mellville, NY 11747
(516) 351-1000
FAX: 516-351-1606

NEW JERSEY (Southern)

Delta Technical Sales, Inc.
122 N. York Road, Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

NEW MEXICO

Quatra Associates
600 Autumwood Place, S. E.
Albuquerque, NM 87123
(505) 296-6781
FAX: 505-292-2092

NEW YORK (Metro)

Parallax
734 Walt Whitman Road
Melville, NY 11747
(516) 351-1000
FAX: 516-351-1606

NEW YORK

Electra Sales Corp.
333 Metro Park
Suite M103
Rochester, NY 14623
(716) 427-7860
FAX: 716-427-0614

Electra Sales Corp.
6700 Old Collamer Rd.
E. Syracuse, NY 13057
(315) 463-1248
FAX: 315-463-1717

NORTH CAROLINA

The Novus Group, Inc.
102L Commonwealth Court
Cary, NC 27511
(919) 460-7771
FAX: 919-460-5703

The Novus Group, Inc.
1775 Cox Rd.
Weddington, NC 28105
(704) 846-4044
FAX: 704-846-4055

NORTH DAKOTA

Com-Tek
6525 City West Parkway
Eden Prairie, MN 55344
(612) 941-7181
FAX: 612-941-4322

OHIO

Bear Marketing, Inc.
3554 Brecksville Road
PO Box 427
Richfield, OH 44286-0427
(216) 659-3131
FAX: 216-659-4823

Bear Marketing, Inc.
240 W. Elmwood Drive
Suite 2010
Centerville, OH 45459-4248
(513) 436-2061
FAX: 513-436-9137

OKLAHOMA

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
FAX: 214-437-0897

OREGON

Thorson Company Northwest
9600 S.W. Oak Street,
Suite 320
Portland, OR 97223
(503) 293-9001
FAX: 503-293-9007

PENNSYLVANIA

Delta Technical Sales, Inc.
122 N. York Rd., Suite 9
Hatboro, PA 19040
(215) 957-0600
FAX: 215-957-0920

Bear Marketing, Inc.
4284 Rt. 8, Suite 211
Allison Park, PA 15101
(412) 492-1150
FAX: 412-492-1155

PUERTO RICO

Semtronic Assoc., Inc.
Mercantile Plaza Building
Suite 816
Hato Rey, PR 00918
(809) 766-0700/0701
FAX: 809-763-8071

RHODE ISLAND

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

SOUTH CAROLINA

The Novus Group, Inc.
102L Commonwealth Court
Cary, NC 27511
(919) 460-7771
FAX: 919-460-5703

SOUTH DAKOTA

Com-Tek
6525 City West Parkway
Eden Prairie, MN 55344
(612) 941-7181
FAX: 612-941-4322

TENNESSEE

The Novus Group, Inc.
366 S. Union Grove Rd.
Friendsville, TN 37737
(615) 995-2365
FAX: 615-995-2215

TEXAS

Bonser-Philhower Sales
8240 MoPac Expwy.,
Suite 135
Austin, TX 78759
(512) 346-9186
FAX: 512-346-2393

Bonser-Philhower Sales
10700 Richmond, Suite 150
Houston, TX 77042
(713) 782-4144
FAX: 713-789-3072

Bonser-Philhower Sales
689 W. Renner Rd., Suite 101
Richardson, TX 75080
(214) 234-8438
FAX: 214-437-0897

TEXAS (El Paso County)

Quatra Associates
600 Autumwood Place SE
Albuquerque, NM 87123
(505) 296-6781
FAX: 505-292-2092

UTAH

Luscombe Engineering Co.
670 East 3900 South #103
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

VERMONT

Genesis Associates
128 Wheeler Road
Burlington, MA 01803
(617) 270-9540
FAX: 617-229-8913

VIRGINIA

Micro Comp, Inc.
8811Timberlake Rd.
Suite 107
Lynchburg, VA 24502
(804) 239-2626
FAX: 804-239-1333

Bear Marketing, Inc.
4284 Rt. 8 Suite 211
Allison Park, PA 15101
(412) 492-1150
FAX:412-492-1155

WASHINGTON

Thorson Company Northwest
12340 NE 8th Street,
Suite 201
Bellevue, WA 98005
(206) 455-9180
FAX: 206-455-9185

WASHINGTON (Vancouver, WA only)

Thorson Company Northwest
9600 S.W. Oak Street
Suite 320
Portland, OR 97223
(503) 293-9001
FAX: 503-993-9007

WASHINGTON D.C.

Micro Comp, Inc.
1421 S. Caton Avenue
Baltimore, MD 21227-1082
(410) 644-5700
FAX: 410-644-5707

WISCONSIN (Western)

Com-Tek
6525 City West Parkway
Eden Prairie, MN 55344
(612) 941-7181
FAX: 612-941-4322

WISCONSIN (Eastern)

Beta Technology Sales, Inc.
9401 N. Beloit, Suite 409
Milwaukee, WI 53227
(414) 543-6609
FAX: 414-543-9288

WYOMING

Luscombe Engineering, Inc.
670 East 3900 South #103
Salt Lake City, UT 84107
(801) 268-3434
FAX: 801-266-9021

Distributed in North America By

Hamilton/Avnet
Locations throughout
the U.S. and Canada.
1-800-888-9236
FAX: 408-743-3003

Marshall Industries
Locations throughout
the U.S. and Canada.
1-800-522-0084
FAX: 818-307-6297

Insight Electronics
Locations throughout
the Western & South
Central U.S.
1-800-677-7716
FAX: 619-587-1380

Phase 1 Technology Corp.
46 Jefryn Blvd.
Deer Park, NY 11729
(516) 254-2600
FAX: 516-254-2693
FAX: 516-254-2695(NY sales)

Nu Horizons
Electronics Corp.
6000 New Horizons Blvd.
Amityville, New York 11701
(516) 226-6000
FAX: 516-226-6262

INTERNATIONAL SALES REPRESENTATIVES

ASEAN

(Singapore, Malaysia, Indonesia, Thailand, Phillippines, Brunei)
MEMEC Asia Pacific Ltd.
10 Anson Rd. #14-02
International Plaza
Singapore 0207
Tel: (65) 222-4962
FAX: (65) 222-4939

AUSTRALIA

Advanced Component Dist.
106 Belmore Rd. North
Riverwood, N.S.W. 2210
Sydney, Australia
Tel: (61) 2-534-6200
FAX: (61) 2-534-4910

Advanced Component Dist.
Unit 2, 17-19 Melrich Road
Bayswater VIC 3153
Melbourne, Australia
Tel: (61) 3-762-7644
FAX: (61) 3-762-5446

Advanced Component Dist.
Enterprise Unit 1, Suite 1
Technology Park
Bentley WA 6102
Australia
Tel: (61) 9-472-3232
FAX: (61) 9-470-2303

Advanced Component Dist.
20D William Street
Norwood SA 5067
Australia
Tel: (61) 8-364-2844
FAX: (61) 8-264-2811

Advanced Component Dist.
Suite 1, 10-48 Beaudesert Rd.
Cooper Plains QLD 4108
Queensland, Australia
Tel: (61) 7-875-1113
FAX: (61) 7-275-3662

AUSTRIA

Eljapex GmbH
Eitnergasse 6
A-1232 Wien
Austria
Tel: (43) 8615-31-3211
FAX: (43) 1-861531201

BELGIUM & LUXEMBURG

Rodelco NV
Limburg Stirum 243
1780 Wemmel
Belgium
Tel: (32) 2-460-0560
FAX: (32) 2-460-0271

CANADA (BRITISH COLUMBIA)

Thorson Company Northwest
12340 N.E. 8th Street
Suite 201
Bellevue, WA 98005 USA
Tel: (206) 455-9180
FAX: (206)-455-9185

Electro Source
6875 Royal Oak Ave.
Burnaby BC V5J 4J3
Canada
Tel: (604) 435-2533
FAX: (604)-435-2538

CANADA (OTTAWA)

Electro Source, Inc.
300 March Road, Suite 203
Kanata, Ontario K2K 2E4
Canada
Tel: (613) 592-3214
FAX: (613)-592-4256

CANADA (TORONTO)

Electro Source, Inc.
230 Galaxy Blvd.
Rexdale Ontario M9W 5R8
Canada
Tel: (416) 675-4490
FAX: (416)-675-6871

CANADA (QUEBEC)

Electro Source
6600 TransCanada Hwy
Suite 420
Point Claire Quebec H9R 4S2
Canada
Tel: (514) 630-7486
FAX: 514-630-7421

CHINA PEOPLE'S REPUBLIC

MEMEC Asia Pacific Ltd.
Unit No.2308-2319, Tower 1
Metroplaza, Hing Fong Rd.
Kwai Fong, N.T. Hong Kong
Tel: (852) 410-2780
FAX: (852) 401-2518

CZECH REPUBLIC

Eljapex/Elbatex GmbH
Prechodni 11
CZ-140 00 Praha 4
Czech Republic
Tel: (02)-692-8087
FAX: (02)-471-82-03

DENMARK

Dana Tech A/S
Krogshoejvej 51
DK-2880 Bagsvaerd
Denmark
Tel: (45) 44-37 71 10
FAX: (45) 44-37 71 12

Dana Tech AS
Egsagervej 8
DK-8230 Rabyhoej
Denmark
Tel: (45) 86-253100
FAX: (45) 86-253102

FINLAND

Field OY Instrumentarium
Niittylänpouku 10
SF-00620 Helsinki
Finland
Tel: (358) 0-7571011
FAX: (358) 0-798853

FRANCE

Rep'tronic
1 Bis, rue Marcel Paul
Bât A
Z.I. La Bonde
F-91300 Massy
France
Tel: (33) 1-60139300
FAX: (33) 1-60139198

AVNET/EMG.
79 Rue Pierre Sémard
92320 Chatillon
France
Tel: (33) 1 49 65 25 00
FAX : (33) 1 49 65 27 38

AVNET Composant
Sud-Ouest
Innopolis Hall A
Voie No. 1-BP 404
31314 Labège Cédex
France
Tel: (33) 61 39 21 12
FAX: (33) 61 39 21 40

AVNET Composant
Aquitaine
16 Rue François Arago
Zi du Phare
33700 Mérignac
France
Tel: (33) 56 55 92 92
FAX: (33) 56 34 39 99

AVNET Composant
Rhône-Auvergne
Parc Cluc du Moulin à Vent
Bât 26
33 Av. du docteur G. Levy
69200 Vénissieux
France
Tel: (33) 78 00 07 26
FAX: (33) 78 01 20 57

AVNET Composant
Provence Côte D'Azur
8300 Toulon
France
Tel: (33) 94 03 32 56
FAX: (33) 94 36 02 15

AVNET Composant
Ouest
Technoparc-Bât.E
4 Av. des Peupliers
35510 Césson Sévigné
France
Tel: (33) 99 83 84 85
FAX: (33) 99 83 80 83

AVNET Composant
Rhône-Alpes
Miniparc - Zac des Béalières
23 Av. de Granier
38240 Meylan
France
Tel: (33) 76 90 11 88
FAX: (33) 76 41 04 09

AVNET Composant
Nantes
Le Sillon de Bretagne
23e étage-Aile C
8 Av.des Thébaudières
44802 Saint Herblain
France
Tel: (33) 40 63 23 00
FAX: (33) 40 63 22 88

AVNET Composant
Est
19 Rue de Rennes, BP 163
54186 Heillecourt Cédex
France
Tel: (33) 83 53 12 34
FAX: (33) 83 56 70 03

GERMANY

Avnet E2000
Stahlgruberring 12
81829 München
Germany
Tel: (49) 89-45110-01
FAX: (49) 89-45110-209

Avnet E2000
Kurfürstenstr. 126
10785 Berlin
Germany
Tel: (49) 30-2110761
FAX: (49) 30-2141728

Avnet E2000
Ivo-Hauptmann-Ring 21
22159 Hamburg
Germany
Tel: (49) 40-645570-0
FAX: (49) 40-6434073

Avnet E2000
Benzstr. 1
70826 Gerlingen
Stuttgart Germany
Tel: (49) 7156-356-0
FAX: (49) 7156-28084

Avnet E2000
Heinrich-Hertz-Str. 52
40699 Erkrath
Düsseldorf Germany
Tel: (49) 211-92003-0
FAX: (49) 211-92003-99

Avnet E2000
Schmidtstr. 49
60326 Frankfurt/M.
Germany
Tel: (49) 69-973804-0
FAX: (49) 211-7380712

Avnet E2000
Kilianstr. 251
90411 Nürnberg
Germany
Tel: (49) 911-995161-0
FAX: (49) 911-515762

Metronik GmbH
Leonhardsweg 2
82008 Unterhaching
München Germany
Tel: (49) 89-611080
FAX: (49) 89-6116468

Metronik GmbH
Zum Lonnenhohl 38
44319 Dortmund
Germany
Tel: (49) 231-2141741/43
FAX: (49) 231-210799

Metronik GmbH
Gottlieb-Daimler-Str.7
24568 Kaltenkirchen
Hamburg Germany
Tel: (49) 41-91-4206
FAX: (49) 41-91-4428

Metronik GmbH
Siemenstr. 4-6
68542 Heddesheim
Mannheim Germany
Tel: (49) 6203-4701/03
FAX: (49) 620345543

Metronik GmbH
Pilotystr. 27/29
90408 Nürnberg
Germany
Tel: (49) 911-544966/68
FAX: (49) 911-542936

Metronik GmbH
Löwenstr. 37
70597 Stuttgart
Germany
Tel: (49) 711-764033/35
FAX: (49) 711-7655181

Metronik GmbH
Liessauer Pfad 17
13503 Berlin
Germany
Tel: (49) 30-436-1134
FAX: (49) 30-431-5956

Metronik Systeme
Grenzstr. 26
06112 Halle
Germany
Tel: (49) 345-823-352
FAX: (49) 345-823-346

Metronik GmbH
Alsfelderstr. 7
64289 Darmstadt
Germany
Tel: (49) 6151-73-0540
FAX: (49) 6151-71-6652

Intercomp
Am Hochwald 42
82319 Starnberg
München Germany
Tel: (49) 8151-16044
FAX: (49) 8151-79270

Intercomp
Meisenweg 19
65527 Niedernhausen
Frankfurt Germany
Tel: (49) 6128-71140
FAX: (49) 6128-72228

GREECE

Peter Caritato & Assoc. S. A.
Llia Iliot, 31
Athens 11743 Greece
Tel: (30) 1-9020115
FAX: (30) 1-9017024

HONG KONG

MEMEC Asia Pacific Ltd.
Unit No. 2308-2319, Tower I
Metroplaza, Hing Fong Road,
Kwai Fong, N.T.
Hong Kong
Tel: (852) 410 2780
FAX: (852) 401 2518

HUNGARY

Dataware KFT/Elbatex GmbH
Angol Utca 22
H-1149 Budapest
Hungary
Tel: (36) 1163 5081
FAX: (36) 1251-5517

Eljapex/Elbatex GmbH
Vaci u 202
H-1138 Budapest
Hungary
Tel: (36) 1-140-9194
FAX: (36) 1-220-9478

INDIA

Core EI Micro Systems
45131 Manzanita Court
Fremont, California 94539
USA
Tel:(510) 770-1066
FAX: 510-657-1525

Core EI Logic System
1181 Fergussion Collage Rd.
Shivaji Nagar
Pune 411005 India
Tel: (91) 212-323982
FAX: (91) 212-323985

CG-CoreEI LogicSystems
Support Centre
EDT Crompton Greaves Ltd.
218/30 39th "A" Cross
9th Main, V Block, Jayanager,
Bangalore 560041 India
Tel: (91) 80-648055
FAX: (91) 1-80-633801

CG-CoreEL LogicSystems
Support Centre
EDT Crompon Greaves Ltd.
Kanjur, Bhandup,
Bombay 400078 India
Tel: (91) 22-5785491
FAX: (91) 22-5782224

IRELAND

Memec Ireland Ltd.
Block H Lock Quay
Clare Street
Limerick
Ireland
Tel: (353) 61-411842
FAX: (353) 61-411888

ISRAEL

E.I.M International Ltd.
9 Hashiloach Street
P.O. Box 4000
Petach Tikva 49130
Israel
Tel: (972) 3-92 33257
FAX: (972) 3-924 4857

ITALY

ACSIS S.R.L.
Via Alberto Mario. 26
20149 Milano, Italy
Tel: (39) 2-48022522
FAX: (39) 2-48012289

Silverstar-Celdis
Viale Fulvio Testi N.280
20126 Milano, Italy
Tel: (39) 2-66125 1
FAX: (39) 2-661014275

Silverstar-Celdis
Via Collamarini, 22
40139 Bologna, Italy
Tel: (39) 51-538500
FAX: (39) 538831

Silverstar-Celdis
Via Paisiello,30
00162 Roma, Italy
Tel: (39) 6-8848841
FAX: (39) 6-8553228

Silverstar-Celdis
Piazza Adriano 9
10139 Torino, Italy
Tel: (39) 11-43-457000
FAX: (39) 114473306

Silverstar-Celdis
Centro Direzionale Benelli
Via M. Del Monaco, 16
6100 Pesaro, Italy
Tel: (39) 721-26560
FAX: (39) 721-400896

Silverstar-Celdis
Via Masaccio, 175
50019 Firenze, Italy
Tel: (39) 55-572418
FAX: (39) 55-579575

Silverstar-Celdis
Piazza Marini 20/10
Lavagna-Genova
Italy
Tel: (39) 185-325325
FAX: (39) 185-303100

JAPAN

Okura & Co., Ltd.
6-12, Ginza Nichome
Chuo-Ku
Tokyo, 104 Japan
Tel: (81) 3-3566-6364
FAX: (81) 3-3566-2887

Fuji Electronics Co., Ltd.
Ochanomizu Center Bldg.
3-2-12 Hongo. Bunkyo-ku
Tokyo, 113 Japan
Tel: (81) 3-3814-1411
FAX: (81) 3-3814-1414

Okura Electronics Co., Ltd.
3-6, Ginza 2-chome,
Chuo-ku,
Tokyo, 104 Japan
Tel: (81) 3-3564-6871
FAX: (81) 3-3564-6870

Okura Electronics
Service Co., Ltd.
Kyoei Bldg.
5-3, Kyobashi 3-chome,
Chuo-ku,
Tokyo, 104 Japan
Tel: (81) 3-3567-6501
FAX: (81) 3-3567-7800

Tokyo Electron Ltd.
P. O. Box 7006
Shinjuku Monolith
3-1 Nishi-Shinjuku 2-chome,
Shinjuku-ku,
Tokyo, 163 Japan
Tel: (81) 3-3340-8193
FAX: (81) 3-3340-8408

Towa Elex Co., Ltd.
Unity Bldg. 5F
6-5 Nihonbashi Tomisawa-cho
Chuo-Ku, Tokyo 103
Japan
Tel: (81) 3-5371-3411
FAX: (81) 3-5371-4760

Varex Co., Ltd.
Nippo Shin-Osaka No. 2 Bldg.
1-8-33, Nishimiyahara,
Yodogawa-ku,
Osaka, 532 Japan
Tel: (81) 6-394-5201
FAX: (81) 6-394-5449

Inoware 21, Inc.
TSI Nihonbashi Hamacho
Daini Bldg.
3-36-5, Nihonbashi Hamacho
Chuo-ku,
Tokyo, 103 Japan
Tel: (81) 3-5695-1521
FAX: (81) 3-5695-1524

KOREA

MEMEC Asia Pacific Ltd.
4FL, Je Woong Bldg.,176-11
Nonhyun-dong
Kangnam-ku
Seoul,135-010, South Korea
Tel: (82) 2-518-8181
FAX: (82) 2-518-9419

THE NETHERLANDS

Rodelco BV
P.O.Box 6824
Takkebijsters 2
4802 HV Breda
The Netherlands
Tel: (31) 76-784911
FAX: (31) 76-710029

NEW ZEALAND

ACD
106 Belmore Rd., North
Riverwood, N.S.W. 2210
Sydney, Australia
Tel: (61) 2-534-6200
FAX: (61) 2-534-4910

NORWAY

BIT Elektronikk AS
Smedsvingen 4
P.O. Box 194
1364 Hvalstad
Norway
Tel: (47) 2-98 13 70
FAX: (47)2-98 13 71

POLAND

Eljapex/Elbatex GmbH
Ul. Hoza 29/31-6
PL-00-521 Warszawa
Poland
Tel: (48) 2625-4877
FAX: (48) 2221-6331

PORTUGAL

Componenta Componentes
Electronicos LDA
R. Luis De Camoes, 128
1300 Lisboa
Portugal
Tel: (351) 1 3621283/4
FAX: (351) 1 3637655

RUSSIA

Scan
10/32 "B" Druzhby St.
117330 Moskva
Russia
Tel: (7) 095-1436641
FAX: (7) 095-9382247

Vostorg
3 Rue des Acacias
91370 Verrieres le Buisson
France
Tel: (33) 1-6920-4613
FAX: (33) 1-6011-5543

SINGAPORE

MEMEC Asia Pacific Ltd.
Singapore Representative Office
10 Anson Road #14-02
International Plaza
Singapore 0207
Tel: (65)-222-4962
FAX: (65)-222-4939

SLOVAK REPUBLIC

Topoicianska 23
SQ-851 05 Bratislava
Tel: (42) 7831-320
FAX: (42) 7831-320

SLOVENIA/CROATIA

Eljapex/Elbatex GmbH
Stegne 19, PO Box 19
SLO-61-117 Ljubljana
Tel: (061) 191-126-507
FAX: (061) 192-398-507

SOUTH AFRICA

South African Micro-Electronic
Systems (PTV) Ltd.
2 Rooibok Avenue
Koedoespoort, Pretoria
Republic of South Africa
Tel: (27) 12-736021
FAX: (27) 12-737084

Interface International Corp.
15466 Los Gatos Blvd.,
Suite 211
Los Gatos, CA 95032 USA
Tel: (408) 356-0216
FAX: (408) 356-0207

SOUTH AMERICA

DTS Ltda.
Rosas 1444
Santiago
Chile
Tel: (56) 2-699-3316
FAX: (56) 2-697-0991

Reycom Electronica SRL
Uruguay 362 Peso 8 - Depto. F
1015 Buenos Aires
Argentina
Tel: (54) 1-304-201-8/9
FAX: (54) 1-304-2010

Hitech
Divisao de Hicad Sistemas Ltda.
Av. Eng. Louiz Carlos Berrini 801
04571-Brooklin
São Paulo, Brazil
Tel: (55) 11-531-9355
FAX: (55) 11-240-2650

SOUTHEAST ASIA

MEMEC Asia Pacific Ltd.
Unit No.2308-2319, Tower I
Metroplaza, Hing Fong Road,
Kwai Fong, N.T.
Hong Kong
Tel: (852) 410-2780
FAX: (852) 401-2518

SPAIN

ADM Electronics SA
Calle Tomas Breton,
no 50, 3-2 Planta
28045 Madrid
Spain
Tel: (34) 91-5304121
FAX: (34) 91-5300164

ADM Electronics SA
Mallorca 1
08014 Barcelona
Spain
Tel: (34) 3-4266892
FAX: (34) 3-4250544

ADM Electronica, SA
Herriro Gudarien, 8-10
48200 Durango (Vizcaya)
Spain
Tel: (34)-4-6201572
FAX: (34)-4- 6202331

SWEDEN

DipCom Electronics AB
P.O. Box 1230
S-164 28 Kista
Sweden
Tel: (46) 8 752 24 80
FAX: (46) 8 751 36 49

SWITZERLAND

Fenner Elektronik AG
Abteilung Bauteile
Gewerbestrasse 10
CH-4450 Sissach
Switzerland
Tel: (41) 61 975 00 00
FAX: (41) 61 971 56 08

TAIWAN

MEMEC Asia Pacific Ltd.
9F-2, No. 225
Nanking E. Rd, Sec. 5
Taipei
Taiwan R.O.C.
Tel: (886) 2-760-2028
FAX: (886) 2-765-1488

Solomon Technology Corp.
5th Floor, No. 293, Sec. 5
Chung Hsiao E. Rd.
Taipei
Taiwan, R.O.C.
Tel: (866) 2-760-5858
FAX: (866) 2-764-7500

UK

Microcall Ltd.
17 Thame Park Road
Thame
Oxon OX9 3XD
England
Tel: (44) 844-261939
FAX: (44) 844-261678

Cedar Technologies
The Old Water Works
Howse Lane
Bicester
Oxfordshire OX6 8XF
England
Tel: (44) 869-322366
FAX: (44) 869-322377

Avnet EMG Ltd.
Jubilee House
Jubilee Road
Letchworth
Hertfordshire SG6 1QH
England
Tel: (44) 462-488500
FAX: (44) 462-488555

The Programmable Logic Company℠

2100 Logic Drive, San Jose, CA 95124-3400
Tel: (408) 559-7778 EasyLink 62916309
TWX: 5106008750 XILINX UQ Fax: (408) 559-7114

0401224

Printed in U.S.A.

PN0401224